Université Henri Poincaré – Nancy I

Minimum Receiving Node and Minimum Energy Broadcast in All-Wireless Networks

MÉMOIRE

5 September 2003

pour l'obtention du

DEA de l'Université Henri Poincaré – Nancy I

(Spécialité Informatique)

par

YIN Fei

Composition du jury

Dominique Méry
Didier Galmiche
Noëlle Carbonell

Encadrant : Johanne Cohen

Laboratoire Lorrain de Recherche en Informatique et ses Applications — UMR 7503

Table des matières

Table des figures

ABSTRACT

An *all-wireless* network is a collection of wireless mobile hosts forming a temporary network without the aid of any established infrastructure or centralized administration. In such an environment, it may be necessary for one mobile host to enlist the aid of other hosts in forwarding a packet to its destination, due to the limited range of each mobile host's wireless transmissions. These hosts are battery operated and hence they need to be energy conserving so that the battery life is maximized. Since batteries provide limited power, a general constraint of wireless communication is the short continuous operation time of mobile terminals. Therefore, power management is one of the most challenging problems in wireless communication.

The energy for transmission of a packet in the *wireless channel* is quite significant and turn out to be the highest energy-consuming component of the wireless device. Hence there is a need for designing minimum energy broadcasting algorithms that ensure a longer battery life. Efficient minimum energy broadcast tree can greatly reduce energy consumption and lead to a longer battery life of the device. Thus, in *all-wireless* networks a crucial problem is to minimize energy consumption in broadcasting environment, include transmission and receiving energy consumption.

In this paper, we study the construction of the source-initiated (one-to-all) wireless broadcast tree to minimize the total required power (transmission power and receiving power) for a given source node, a group of intended destination nodes and a given propagation constant in an *all-wireless* environment. We introduce and evaluate algorithms for broadcast tree construction in un-infrastructure, *all-wireless* applications. We first give an introduction on the problem of power-optimal in unicast environment, and give a new heuristic : Minimum Receiving Node Shortest Path Tree(**MRN-SPT**), then we apply our conception in broadcast environment, which the broadcast nature of the radio transmission can be exploited to optimize energy consumption.

Several works [10, 16, 17] have given their algorithms for the transmission energy efficiency broadcast in *all-wireless* networks, but them do not consider the receiving energy consumption. Base on their algorithms, we then describe two new heuristics : Minimum Receiving Node Embedded Wireless Multicast Advantage (**MRN-EWMA**) and Minimum Receiving Node Broadcast Incremental Power (**MRN-BIP**) in broadcast environment, which fully utilize the *wireless multicast advantage*[17], consider both the transmission and receiving energy consumption. We will demonstrate they have the better performance compared with the original proposals.

KEYWORDS

All-wireless network, wireless ad-hoc network, multi-hop, MANET, minimum-energy networks, energy efficiency, wireless multicast advantage, receiving node, transmission range

1 INTRODUCTION

In recent years, the *all-wireless* applications have enjoyed a tremendous and rapid expansion in civil and military domains. The continued miniaturization and the extraordinary rise of processing power available in mobile devices combine to put more and better computer-based *all-wireless* applications into the hands of a growing segment of the population. The rise of wireless application will change what it means to be "in touch" : already many people use their wired backbone networks for taking and transmitting messages while they are away and rely on their their mobile devices for more important or timely messages. It is an indication that the *all-wireless* networks are expected to fulfill a critical role in applications in which wired backbone networks are not available or not economical to build.

An *all-wireless* network consists of numerous devices that are equipped with processing, memory and wireless communication capabilities, and are linked via short-range ad hoc radio connections. This kind of network has no pre-installed infrastructure, but all communication is supported by *multi-hop* transmissions, which means that the intermediate node transmit packet between communicating parties. It provide the only solution in situations where instant infrastructure is needed and no central system backbone and administration exist. For each node in *all-wireless* network equipped with a limited power source (battery) and operates unattended, a general constraint of wireless communication is the short continuous operation time of mobile terminal. Therefore, power management is one of the most challenging problems in wireless communication, and recent researchs[1, 4, 6, 8, 9, 10, 11, 13, 16, 17, 18] have addressed this topic.

In this introductory section, we consider some general topics that provide context for the rest of the sections in this paper. In the next subsection, we describe a general introduction of the construction of ad hoc network and some of the properties which affecting the design decisions that various approaches have taken (section 1.1). In section 1.2, we introduce the energy consumption in ad hoc networks, explain why and how we should minimize the energy consumption in the application of ad hoc network, finally give the operation wireless communication model. In section 1.3, the minimum transmission energy broadcast problem is introduced. Based on it, we give the minimum transmission and receiving energy broadcast problem. After that we show several relative algorithms about constructing the minimum transmission energy broadcasting tree in Section 1.4. In Section 2, we afford our conception for constructing the minimum receive energy consumption shortest path tree in unicast environment, give the pseudo code and the proof. Then in section 3, we apply this conception on the construction of the minimum receiving node and minimum transmission energy broadcast tree in multicast and broadcast environment and give two algorithms. In Section 4, we give the performance evaluation of our algorithms and we finally get the conclusion in Section 5.

1.1 Ad Hoc Networks

A Mobile Ad hoc NETwork (**MANET**)[1] also termed mobile multi-hop radio, peer-to-peer, or *all-wireless* networks, is a network in which a set of mobiles cooperatively maintain network connectivity. The term **MANET** describes distributed, mobile, wireless, multihop networks that operate without the benefit of any existing infrastructure except for the nodes themselves. Roots of this technology could be traced back to the early 1970s with the **DARPA**(Defense Advanced Research Projects Agency) **PRNet** (multi-hop multiple-access Packet Radio NETwork) and the **SURAN** (SUrvivable RAdio Networks) projects. Although this network concept has been originally considered in the context of packet radio networks earlier, it has become very popular again during the past few years. The work is going on within the IETF's MANET work group for standards and the research is very active throughout the world.

The **MANET** is an on-demand network architecture which is completely un-tethered from physical wires. An example of an ad hoc network is shown in figure 1, illustrating a collection of nine nodes along

[1] Working group established by Internet Engineering Task Force(**IETF**) in 1997, *http ://www.ietf.org/html.charters/manet-charter*

with the links between them.

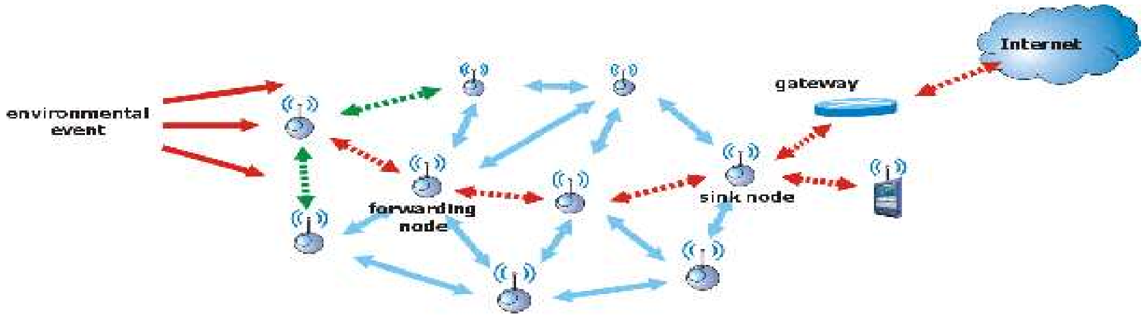


FIG. 1 – Ad hoc wireless network architecture

Ad hoc networks are characterized by dynamic, unpredictable, random, *multi-hop* topologies with typically no infrastructure support :

- It can be rapidly deployed without relying on pre-existing fixed network infrastructure ;
- The nodes can dynamically join and leave the network, frequently, often without warning, and possibly without disruption to other nodes' communication ;
- The nodes can also be highly mobile, thus rapidly changing the node constellation and the presence or absence of links.

Such a network supports "anytime" and "anywhere" computing, allowing the spontaneous formation and deformation of mobile networks.

From a graph theoretic point of view, an ad hoc network is a graph, $G=(V, E(t))$, which is formed by denoting each of the V mobile hosts by a node and drawing an edge between two nodes if they are in direct communication range of each other. The set of edges, $E(t)$, so formed is a function of time and keeps changes as nodes in the ad hoc network move around.

In **MANET**, whether the range of wireless transmission should be large or small compared to the geographic distribution of the mobile wireless nodes. If all of the wireless nodes are within the *transmission range* of each other, no routing is needed (nodes communicate with each other through a single-hop transmission if the destination is within the transmission node's transmission range), and the ad hoc network is fully connected. While if some of the wireless nodes are not within the *transmission range* of each other, combined with the lack of infrastructure routers, the restricted range of wireless transmission indicates the need for *multi-hop* routing(using intermediate nodes to transmit the message).

Nodes in the **MANET** exhibit nomadic behavior by freely migrating within some area, dynamically creating and tearing down associations with other nodes. Groups of nodes that have a common goal can create formations (clusters) and migrate together. Nodes can communicate with each other at any time and without restrictions, except for connectivity limitations and subject to security provisions.

MANETs are intended to provide a data network that is immediately deployable in arbitrary communication environment and is responsive to changes in network topology. Because ad hoc networks are intended to be deployable anywhere, existing infrastructure may not be present. The mobile nodes are thus likely to be the sole elements of the network. Differing mobility patterns and radio propagation conditions that vary with time and position can result in intermittent and sporadic connectivity between adjacent nodes. The result is a time-varying network topology.

Ad hoc networks are helpful in situations in which temporary network connectivity is needed, and are often used for military environments, disaster relief, and so on. Examples of the use of the **MANET**s[14, p. 222] are :

- **tactical operation** : for fast establishment of military communication during the deployment of forces in unknown and hostile terrain ;
- **rescue missions** : for communication in areas without adequate wireless coverage ;
- **national security** : for communication in times of national crisis, where the existing communication infrastructure is non-operational due to a natural disaster or a global war ;
- **law enforcement** : for fast establishment of communication infrastructure during law enforcement operations ;
- **commercial use** : for setting up communication in exhibitions, conferences, or sales presentations ;
- **education** : for operation of wall-free (virtual) classrooms ;
- **sensor networks** : for communication between intelligent sensors mounted on mobile platforms.

Although the *all-wireless* network have many advantages compared with wired network in the situation where the temporary network connectivity is needed, it also face one rigorous challenge : the limited power supply of mobile units. In the next subsection, we will introduce the energy consumption in the ad hoc network and give the operation wireless communication model.

1.2 Operation Model

Power-efficient networks are currently being extremely popular within the ad hoc networking research. The motivation for power-efficient thinking for wireless communications is obvious, as summarized in :

- **Functional utility** : New features and functionality usually costs additional energy. By increasing energy efficiency, devices may meet new use demands without reduced useful lifetime.
- **Size and weight** : Increased power efficiency can allow smaller and lighter power source.
- **Maintenance** : Power sources will always need to be replaced or re-charged at some point, and the cost for this can vary from inconvenient to prohibitive.
- **Environmental** : Battery designs contain acids and heavy metals, which must be disposed of properly.

The source of power consumption, with regard to network operation, can be communication related and computation related. In this paper, we only consider the communication related power consumption. Communication involves usage of the transceiver at the source, intermediate, and destination nodes. The transmitter is used for sending control, route request and response, as well as data packets originating at or routed through the transmitting node. The receiver is used to receive data and control packets-some of which are destined for the receiving node and some of which are forwarded. A typical mobile radio may exist in four modes : transmit, receive , idle and standby. Maximum power is consumed in the transmit mode, and the least in the standby mode in which the interface can neither transmit nor receive. To be able to transmit or receive, an interface must explicitly transition to the idle state, which requires both time and energy. In addition, turnaround between transmit and receive modes (and vice-versa) typically takes between 6 and 30 microseconds. There are several examples about the four modes power consumption[5, 8] :

Table 1.1 : Power consumption for four modes

	transmit	**receive**	**idle**	**standby**
Aironet PC4800	1.4-1.9W	1.3-1.4W	1.34W	0.075W
Lucent Bronze	1.3W	0.97W	0.84W	0.066W
Lucent Silver	1..3W	0.9W	0.74W	0.048W
Cabletron Roamabout	1.4W	1.0W	0.83W	0.13W

There are two ways that ad hoc network nodes operate on battery power to do communication. First, they might transmit data to a desired recipient. This use of battery power is not part of the overhead of ad hoc networking. Second, a mobile node might offer itself as an intermediate forwarding node for data going between two other nodes in the network. Providing such a service is likely to be costly in terms of power consumption, but without the availability of such forwarding nodes there can be no ad hoc network.

As discussed in the last subsection, nodes in the ad hoc network communicate each other can with single-hop or multi-hop. So if the network is dense enough, only a subset of nodes is required to be relaying nodes to maintain full connectivity. This means that some of the nodes can be put to a standby state only to wake up periodically to see whether there are incoming traffic directly to them. Active nodes form a forwarding backbone in the network. Thus the goal of energy efficient for environments with limited power resources is to optimize the transceiver usage for a given communication task.

In this paper, we consider not only the energy used for transmission but also the energy associated with reception, but neglecting for the present the energy associated with signal processing which is computing related. We give a wireless communication model and then, develop a graph model, which will be used develop the algorithms.

We base our work on the so called *node-based* multicast model[16, 17]. Following similar hypotheses in [17], we assume that :

- **the node locations are fixed and the channel conditions are unchanged**. However, the situation with the mobility of the nodes in the construction of broadcast tree can be addressed by adjusting the transmission power to accommodate the new location of the nodes.
- **sufficient bandwidth resources are available at each node**.
- **the power level of a transmission can be chosen within a given range of values**. Therefore, there is a trade-off between reaching more nodes(more receiving energy consumption) in a single hop using higher transmission power and reaching fewer nodes(fewer receiving energy consumption) using lower transmission power. This trade-off is possible due to the *broadcast nature* of the wireless channel.
- **sufficient transceiver resources are available at each of the nodes in the network**. thus, calls are never blocked because of the unavailability of either a transmitter or receiver.
- **nodes in a network are equipped with omnidirectional antennas**. Thus by a single transmission of a transmitting node, due to the *broadcast nature* of wireless channels, all nodes that fall in the transmission range of the transmitting node can receive its transmission. This property of wireless media is called *Wireless Multicast Advantage*, which we refer to as WMA[17].

We consider source-initiated, circuit-switched, multicast sessions. The network consists of N nodes, which are randomly distributed over a specified region : Each node has several transceivers, and can thus support several multicast sessions simultaneously. Any node is permitted to initiate multicast sessions. Multicast requests and session duration are generated randomly at the network nodes. Each multicast group consists of the source node plus at least one destination node. Additional nodes may be needed as transmission nodes either to provide connectivity to all members of the multicast group or to reduce overall energy consumption or both. The set of nodes that support a multicast session(the source node, all destination nodes, and all transmission nodes) is referred to as a *multicast tree*.

The connectivity of the network depends on the transmission power. We assume that each node can choose its power level, not to exceed some maximum value p_{max} and not to lower some minimum value p_{min}. The nodes in any particular multicast tree do not necessarily have to use the same power levels; moreover, a node may use different power levels for the various multicast trees in which it participates.

Let P denote the set of power levels at which a node can transmit[2]. When a node i transmits at some power level $p \in P$, we assign it a weight, which we call a node's *transmission power*, that is p. The

[2]We assume the cardinality of P to be finite; this does not reduce the generality of our approach, as this cardinality can be arbitrarily large.

connectivity of the network depends on the transmission power. Node i is said to be *connected* to node j if node j falls in the transmission range of node i. This link is then assigned a *link cost* c_{ij}, which is equal to the minimum power that is necessary to sustain link(i,j). According to this *link cost* c_{ij}, we can get the nodes which fall in the transmission range rooted at node i with *link cost* c_{ij}. With each *link cost* c_{ij}, there is a set of nodes which are fall in the transmission range of the transmission node, we call these node as *receiving nodes* of node i.

Next we give a graph model for wireless networks that captures important properties of wireless media (including the Wireless Multicast Advantage). An *all-wireless* network can be modeled by a directed graph $G=(V, E)$, where V represents the finite set of nodes and E the set of communication links between the nodes. Each edge $(i, j) \in E$ has link cost $c_{ij} \in \Re_+$ assigned to it[3], while each node $i \in V$ is assigned a *variable transmission power* p_i^v . The variable transmission power takes a value from the set P defined above. Initially, the variable transmission power assigned to a node is equal to zero, and is set to value $p \in P$ after the node has transmitted at p. Let V_i denote the set of *neighbors* of node i. Node j is said to be a *neighbors* of node i if node j falls in the maximum transmission range of node i, which is determined by p_{max}. All nodes $j \in V_i$ that satisfy $c_{ij} \leq p_i^v$ are said to be *covered* by node i. Thus if node i transmits at power p_{max}, all the nodes from V_i will be covered.

Now we have the operation model, we will next introduce the minimum energy broadcast problem and combine it to explicate our conception.

1.3 Minimum Energy Broadcast Problem

Broadcasting is a communication paradigm that allows to send data packets from a source to multiple receivers. Broadcasting and multicasting in *wireless ad hoc networks* are critical mechanisms in various applications such as information diffusion, sensor networks, and also for maintaining consistent global network information. Broadcasting is more efficient that sending multiple copies the same packet through unicast. *Ad hoc wireless networks* are energy limited because the nodes are usually battery-powered. Therefore, it is highly important to use power-efficient broadcast algorithms for such networks.

Multicasting and broadcasting in ad hoc networks is more challenging than in the wired network, because of the need to optimize the use of several resource simultaneously. Firstly, nodes in ad hoc networks are battery-power limited. Furthermore, data travels over the air and wireless resources are scarce. Secondly, there is no centralized access point or existing infrastructure to keep track of the node mobility. Thirdly, the status of communication links between transmission nodes are a function of their positions, transmission power levels, etc. The mobility of transmission nodes and randomness of other connectivity factors lead to a network with a potentially unpredictable and rapidly changing topology. This mean that by the time a reasonable amount of information about the topology of the network is collected and a tree is computed, there may be very little time before the computed tree becomes useless.

We study source-initiated broadcasting (one-to-all) and multicasting (one-to-many) of "session" traffic. In either case, our objective is to form a minimum-energy(involve minimum transmission and minimum receiving energy consumption) tree, rooted at the source, that reaches all of the desired destinations. The problem of minimum-energy broadcast can exist in two different graph models, namely a *general graph* and a *graph in Euclidean metric space*[16]. In general graph, links are arbitrarily distributed, and have arbitrarily weights chosen from the set P. This graph model is well suited for modeling wireless networks in indoor environments. On the other hand, for graphs in Euclidean metric space, the existence and the weight of the link between two nodes depends exclusively on the distance between the nodes and their transmission levels. This graph model fits well for outdoor scenarios.

[3] We designate with $\Re_+$ strictly positive reals.

The broadcast communication is an important mechanism to communicate information in *all-wireless* networks : when omni-directional antennas are used, every transmission by a node can be received by all nodes that lie within its communication range. Because the **MANET** can be regarded as a distributed system (distributed hardware + distributed control + distributed data), where broadcast is an important communication primitive. In addition, many routing protocols for wireless ad-hoc networks need a broadcast mechanism to update their states and maintain the routes between nodes.

Nodes belonging to a broadcast tree can be divided into two categories : *transmission* nodes and *leaf* nodes. The transmission nodes are those that transmit data to other nodes (transmission node or leaf node), while leaf nodes only receive data. Each node can transmit at different power levels and thus reach a different number of neighboring nodes. Given the source node s, we want to find a set consisting of pairs of transmission nodes and their respective transmission levels so that all nodes in the network receive a message sent by s, and the total transmission and receiving energy expenditure for this task is minimized. We call this broadcasting problem the *minimum receiving node and minimum energy broadcast problem.*

We have seen that the problem of energy efficient broadcast is very important in *all-wireless* network. And the recent works have pay much attention on this topic, and we will introduce three recent algorithms about constructing the minimum transmission energy broadcast tree in ad hoc networks. Base on them, we will afford our conception.

1.4 Relative Algorithms

The problem of minimizing the transmission power consumption of *all-wireless* networks has received significant attention over the last few years[17, 16, 10]. In these works they have introduced the node-based[17] multicast models and upon which they have built several broadcast and multicast heuristics. In the following section, we describe these algorithm respectively.

Embedded Wireless Multicast Advantage

Embedded Wireless Multicast Advantage(EWMA) is supported by Čagalj et al[16], the main objective of EWMA is to construct a minimum transmission energy broadcast tree rooted at the source node. It begin with a feasible solution(an initial feasible broadcast tree–Minimum Spanning Tree) for a given network. For only the transmission nodes have the transmission energy consumption in the broadcast tree, then it improve the solution by exchanging some existing branches in the initial tree for new branches to decrease the number of transmission nodes, that means that to decrease the total energy necessary to maintain the broadcast tree. Thus, after some repeat steps of EWMA, the remain transmission nodes can additionally increase their transmission power to reach more nodes but decrease the number of transmission nodes to lower the total transmission power consumption.

In EWMA, the author defined several variable :
- *gain* : the decrease in the total transmission energies of a broadcast tree obtained by excluding some of the nodes from the set of transmission nodes in MST, in exchange for the increase in transmission power.
- set C : the set of covered nodes in a network.
- set F : the set of transmitting nodes of the final broadcast tree.
- set E : the set of *excluded* node. Node i is said to be an *excluded* node if node i is the transmission node in the initial solution but is not the transmission node in the final solution.

We introduce this algorithm by a example of 10 nodes network(Figure 2). After the MST(Minimum Spanning Tree) has been built in the initialization, we know which nodes in the MST are transmission nodes, and their respective transmission power. In our example the source node is node 10 and the transmission nodes are 1, 2, 6, 8, 9, 10 and their transmission power are 5, 8, 2, 9, 8 and 13, respectively. So the total transmission energy of MST is e_{MST}=5+8+2+9+8+13=45.

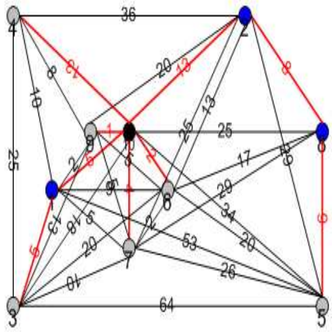


FIG. 2 – The network example and its MST(e_{MST}=45)

Initially, EWMA determine the respective gain of the nodes in the set C-F-E one by one to build a broadcast tree.

In the example, C=$\{10\}$, F=E=$\{\emptyset\}$, we get the set C-F-E contains just the source node 10. So, we determine the maximum gain of node 10 when exclude the other transmission nodes in the initial MST. Because the initial transmission power of node 10 is e_{10}=13, so nodes 1, 2, 4, 6, 7 and 9 are included in the transmission range of node 10. Exactly the node 6 and node 9 are no more the transmission nodes. Thus in the example, there are only three transmission nodes we can try to exclude : nodes 1, 2 and 8.

See Figure 2, in order to exclude node 1 :

source node 10 has to increase its transmission power : Δe^1_{10}=$\omega_{10,3}$-e_{10}=18-13=5
the gain for exclude node 1 (g^1_{10}) : g^1_{10}=e_1-Δe^1_{10}=5-5=0

where $\omega_{10,3}$ is the transmission power from source node 10 to node 3, e_1 is the energy at which node 1 transmits in MST.

In order to exclude node 2 :

the source node 10 has to increase its transmission power : Δe^2_{10}=$\omega_{10,8}$-e_{10}=25-13=12
the gain of excluding node 2 (g^2_{10}) : g^2_{10}=e_1+e_2-Δe^2_{10}=5+8-12=1

where $\omega_{10,8}$ is the transmission power from source node 10 to node 8, e_1 and e_2 is the respective energy at which node 1 and node 2 transmit in MST. With the transmission power of node 10 increase to e_{10}=$\omega_{10,8}$=25, the nodes 3 and 8 will fall in the transmission range of node 10, so the node 1 and 2 will no more be transmission nodes.

In order to exclude node 8 :

the source node 10 has to increase its transmission power : Δe^8_{10}=$\omega_{10,5}$-e_{10}=34-13=21
the gain of excluding node 8 (g^8_{10}) : g^8_{10}=e_1+e_2+e_8-Δe^8_{10}=5+8+9-21=1

where $\omega_{10,5}$ is the transmission power from source node 10 to node 5, e_1, e_2 and e_8 are the respective energy at which node 1, node 2 and node 8 transmit in MST. With the transmission power of node 10 increase to e_{10}=$\omega_{10,5}$=34, the node 3, 5 and 8 will fall in the transmission range of node 10, so the nodes 1, 2 and 8 will no more be transmission nodes.

Second, have the gains for all nodes from C-F-E, the EWMA do one of the following steps :
– EWMA selects a node with the highest positive gain to add into the set F.
– If more than one node have the same highest positive gain, selection will be random.

– If none has the positive gain, EWMA selects the node that includes its *child* nodes[4] in MST at minimum energy.

After that, EWMA adds all the nodes that this node excludes to the set E.

In the example, node 10 is the only one in the set C-F-E. Exclusions of node 2 and node 8 have the same highest positive gain, we select randomly to exclude node 2. Thus the source node 10 is selected to add in the set F to transmit with energy that maximizes its gain, that is $e_{10}=\omega_{10,8}=25$ at which it can cover nodes 1, 2 and all their *child* nodes in MST. Hence, we get $C=\{1, 2, 3, 4, 6, 7, 8, 9, 10\}$, $E=\{1, 2\}$ and $F=\{10\}$.

Third, repeat the above procedure until all nodes in the network are covered.

In our example there still one node not to be covered : node 5. Again EWMA scans the set C-F-$E=\{3, 4, 6, 7, 8, 9\}$ and select node 8 to be the next forwarding node. When node 8 transmits with power $e_8=9$, all nodes are covered($C=\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$) and the algorithm terminate.

Finally, we get the final minimum transmission energy broadcast tree, which the sum of the transmission powers at each of the transmission nodes consisted the total power required to maintain this tree.

In our example, we get $E=\{1, 2\}$, $F=\{8, 10\}$. The result tree, shown in Figure 3, has a total transmission power :

$$e_{EWMA}=e_8+e_{10}=9+25=34.$$

We also can get the total receiving node of the final broadcast tree :

$$\beta_{EWMA}=\beta_8+\beta_{10}=2+8=10.$$

where β_8 and β_{10} are respectively the number of receiving node of node 8 and node 10 when they transmit with their transmission powers in EWMA.

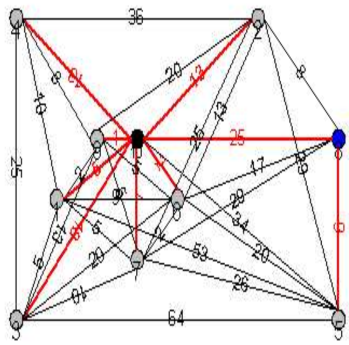


FIG. 3 – The broadcast tree obtained by EWMA algorithm(e_{EWMA}=34)

Broadcast Incremental Power

Broadcast Incremental Power(BIP) is introduced by Wieselthier et al. [17] which is to construct a minimum transmission energy broadcast tree rooted at the source node. It constructs the tree by first determining the node that the source can reach with minimum expenditure of power. After the first node has been added to the tree, BIP continues by determining which uncovered node can be added to the tree at *minimum additional cost*. Thus at some iteration of BIP, the nodes that have already included some

[4]Node j is said to be a *child* node of node i if node j is included in a broadcast tree by node i

node in the tree can additionally increase their transmission power to reach some other yet uncovered node.

We describe the BIP by presenting a simple example of tree construction. Figure 4 show a 10 nodes network with edge power, in which node 10 is the source node.

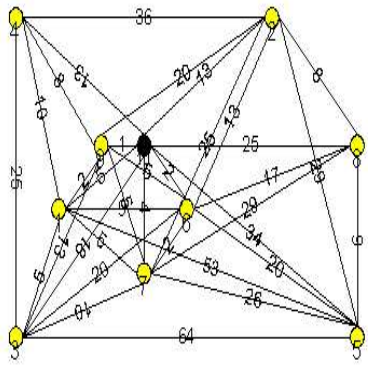


FIG. 4 – The example for algorithm BIP

Initially, the algorithm BIP begin by determining the node that the source can reach with minimum expenditure of power.

In the example, the tree only contain the source node 10, and node 9 is the nearest neighbor of node 10. So node 9 is added to the tree, and the transmission power of node 10 will be e_{10}=$\omega_{10,9}$=1. At this point, two nodes are included in the tree : namely node 10 and node 9(Figure 5).

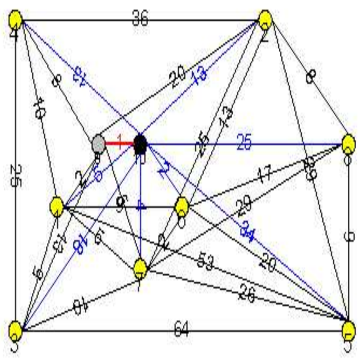


FIG. 5 – The example for algorithm BIP-step1 : 10→9

Second, we then determine which new node can be added to the tree at *minimum additional cost.*

There are two alternatives : node 10 can increase its transmission power to reach a second node, or node 9 can transmit to its nearest neighbor that is not already in the tree. In this example :

node 10 reach node 6 with its minimum additional cost : $min\Delta e_{10}$=$\omega_{10,6}$-e_{10}=1
node 9 reach its nearest neighbor with its minimum additional cost : $min\Delta e_9$=$\omega_{9,1}$=2

where e_{10} is the transmission energy of node 10, $\omega_{10,6}$ is the weight of the edge(10,6), $min\Delta e_{10}$ is the minimum additional cost of node 10. $\omega_{9,1}$ is the weight of the edge(9,1), $min\Delta e_9$ is the minimum additional cost of node 9.

We add node 6 to the tree(Figure 6), and the transmission power of node 10 will be e_{10}=$\omega_{10,6}$=2.

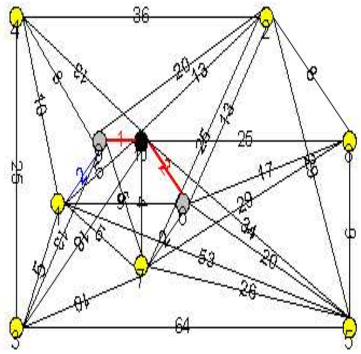


FIG. 6 – The example for algorithm BIP-step2 : 10→6

Third, for each of nodes already in the tree, determine the incremental cost to reach a new node.

There are now three nodes in the tree, namely node 10, node 9 and node 6. For each of these nodes, we determine the incremental cost to reach a new node. Since node 6 and node 9 were not previously transmitting, their respective incremental costs will equal their full transmission powers if they are chosen to transmit, but for node 10, its incremental cost is only the required increase in its transmission power.

In this example :

node 10 reach node 7 with its minimum additional cost :	$min\Delta e_{10}$=$\omega_{10,7}$-e_{10}=2
node 9 reach its nearest neighbor with its minimum additional cost :	$min\Delta e_9$=$\omega_{9,1}$=2
node 6 reach its nearest neighbor with its minimum additional cost :	$min\Delta e_6$=$\omega_{6,7}$=2

We select randomly node 7 to add to the tree(Figure 7), and the transmission power of node 6 will be e_6=$\omega_{6,7}$=2

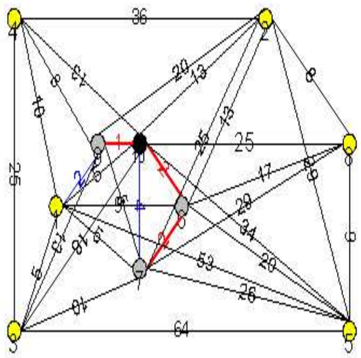
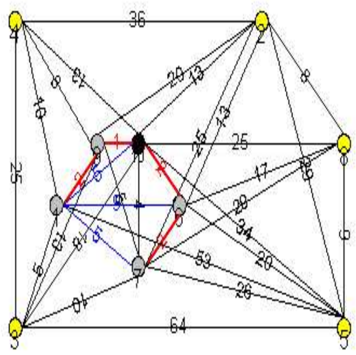


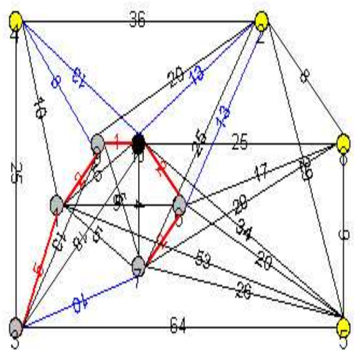
FIG. 7 – The example for algorithm BIP-step3 : 6→7

Continue, we continue this procedure until all nodes are included in the tree.

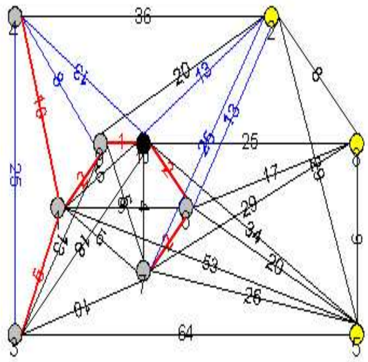
As shown in form Figure 8, the order in which the nodes were added in steps 4 through 9 is : 9 → 1, 1 → 3, 1 → 4, 6 → 2, 6 → 8, 6 → 5. The algorithm BIP terminated.



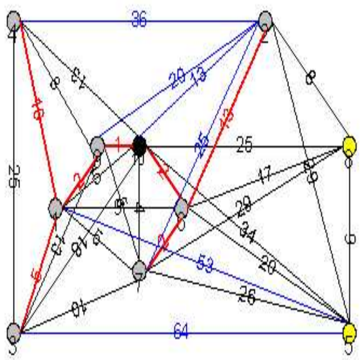
(a) Step 4 : 9→1



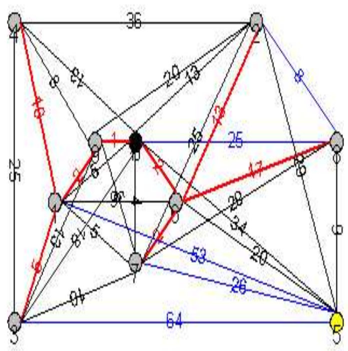
(b) Step 5 : 1→3



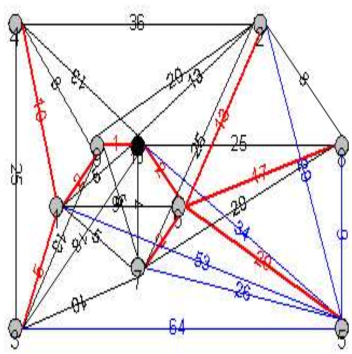
(c) Step 6 : 1→4



(d) Step 7 : 6→2



(e) Step 8 : 6→8



(f) Step 9 : 6→5

FIG. 8 – The example for algorithm BIP-step4-9

Finally, we get the final broadcast tree, as shown in Figure 9. The transmission nodes are nodes 1, 6, 9 and 10. Their transmission power are respective e_1=10, e_6=20, e_9=2, e_{10}=2. So the total power required to maintain this tree is the sum of the transmission powers at each of the transmission nodes, which we can get :

$$e_{BIP}=e_1+e_6+e_9+e_{10}=10+20+2+2=34,$$

and the number of receiving node is :

$$\beta_{BIP}=\beta_1+\beta_6+\beta_9+\beta_{10}=6+6+2+2=16$$

Mini_Broad_Tree

The algorithm Mini_Broad_Tree is supported by Liang[10], it is an approximate algorithm for the problem of minimum transmission energy broadcast tree. It first reduce the problem to an optimization problem on an auxiliary graph and construct an auxiliary graph $G(V, E, \omega_1)$. Then, solve the optimization problem on the auxiliary graph. Finally, the approximate solution for the auxiliary graph gives an approximate solution for the original problem.

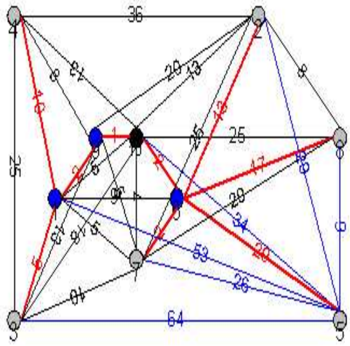


FIG. 9 – Final tree obtained by algorithm BIP–e_{BIP}=34

This algorithm use a approximation algorithm[2] to find an approximate Steiner tree T_{app} on the auxiliary graph. Then it modify the obtained approximate Steiner tree T_{app} by considering the *WMA* property to construct a new tree $T^{'}_{app}$. According to the information given by the tree $T^{'}_{app}$, set the transmission power of each node in the tree, and finally get the minimum-energy broadcast tree.

The algorithm in [2] is an approximation sequence algorithms for Directed Steiner Problems : given a directed graph $G=(V,E)$, a specified root $r \in V$, and a set of terminals $X \subset V(|X|=k)$, the objective is to find the minimum cost arborescence rooted at r and spanning all the vertices in X. And in [2], it named this sequence of algorithms as $A_i(k, r, X)$.

In [2], the author has definite several variables :

- $T_i(k^{'}, v, X)$: the tree returned by $A_i(k, r, X)$
- (u,v) : the edge from u to v in the graph G.
- $c(e)$: the cost of edge e
- $c(T)$: the cost of a tree T
- $k(T)$: the number of terminal in a tree T
- $d(T)$: the density of tree T, $d(T)= \frac{c(T)}{k(T)}$

For the sequence algorithms, $A_1(k, r, X)$, it find the k terminals which are closest to the root and connects them to the root using shortest paths; $A_i(k, r, X)$ repeatedly finds a vertex v and a number $k^{'}$, $1 \leq k^{'} \leq k$ such that the density of the tree $T_{i-1}(k^{'}, v, X) \cup \{(r,v)\}$ is the least among all trees of this form.

We here introduce this algorithm with a 10 nodes network in which node 10 is the source node, shown in Figure 10.

First, using the algorithm in [2] to construct a approximate Steiner tree.

The Steiner tree first only contain the source node 10, we apply the algorithm $A_1(k, r, X)$ on source node 10, we can get 9 nodes are closest to the node 10 and connects them to root. Then we apply the algorithm $A_i(k, r, X)$ on each node of these 9 nodes :

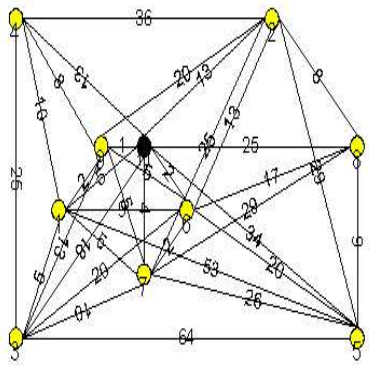


FIG. 10 – The example for algorithm MBT

Nodes	$c(T)$	$k(T)$	$d(T)=\frac{c(T)}{k(T)}$
1	$\omega_{10,1}+(\omega_{1,3}+\omega_{1,4}+\omega_{1,5}+\omega_{1,6}+\omega_{1,7}+\omega_{1,9})=85$	7	12.1428
2	$\omega_{10,2}+(\omega_{2,4}+\omega_{2,5}+\omega_{2,6}+\omega_{2,7}+\omega_{2,8}+\omega_{2,9})=144$	7	20.5714
3	$\omega_{10,3}+(\omega_{3,1}+\omega_{3,4}+\omega_{3,5}+\omega_{3,6}+\omega_{3,7}+\omega_{3,9})=155$	7	22.1428
4	$\omega_{10,4}+(\omega_{4,1}+\omega_{4,2}+\omega_{4,3}+\omega_{4,9})=92$	5	18.4
5	$\omega_{10,5}+(\omega_{5,1}+\omega_{5,2}+\omega_{5,3}+\omega_{5,8}+\omega_{5,6}+\omega_{5,7})=235$	7	33.5714
6	$\omega_{10,6}+(\omega_{6,1}+\omega_{6,2}+\omega_{6,3}+\omega_{6,5}+\omega_{6,7}+\omega_{6,8}\omega_{6,9})=86$	8	10.75
7	$\omega_{10,7}+(\omega_{7,1}+\omega_{7,2}+\omega_{7,3}+\omega_{7,5}+\omega_{7,6}+\omega_{7,8}+\omega_{7,9})=106$	8	13.25
8	$\omega_{10,8}+(\omega_{8,2}+\omega_{8,6}+\omega_{8,5}+\omega_{8,7})=88$	5	17.6
9	$\omega_{10,9}+(\omega_{9,1}+\omega_{9,2}+\omega_{9,3}+\omega_{9,4}+\omega_{9,6}+\omega_{9,7})=54$	7	7.7142

We can see that the tree consist of node 10, node 9 and the child nodes of node9 has the least density among all the trees above. We add node 9 and its child nodes to the Steiner tree, so at this points, there are 8 nodes in the Steiner tree, they are node 1, 2, 3, 4, 6, 7, 9 and source node 10, see Figure 11. There are two nodes uncovered by the tree : node 5 and node 8.

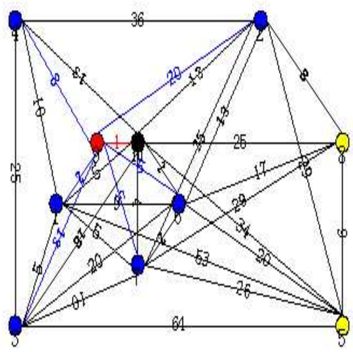


FIG. 11 – The example for algorithm MBT step1

We repeat the algorithm $A_i(k, r, X)$ on the node 5 and node 8 :

Nodes	$c(T)$	$k(T)$	$d(T)=\frac{c(T)}{k(T)}$
5	$\omega_{10,5}+\omega_{5,8}=43$	2	21.5
8	$\omega_{10,8}+\omega_{8,5}=34$	2	17

From the tree above, we can see that the tree constructed by node 10, node 8 and its child nodes has the less density. We add node 8 and its child nodes to the Steiner tree, then all the nodes are covered in the tree. We get the final approximate Steiner tree, see Figure 12.

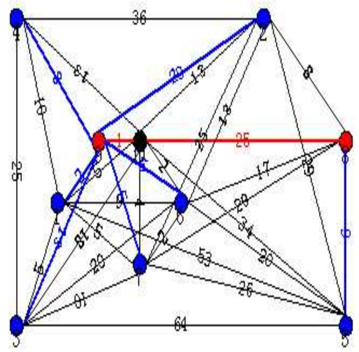


FIG. 12 – The example for algorithm MBT final Steiner tree

Second, we take into account the *WMA* property to modify the obtained tree.

Node 10 can reach node 8 with transmission power $\omega_{10,8}$=25, and also reach node 9 with transmission power $\omega_{10,9}$=1, we can get the transmission power of node 10 should be e_{10}=25. So the nodes 1, 2, 3, 4, 6, 7 and 9 fall in the transmission range of node 10 with transmission power 25. We modify the obtained Steiner tree by replacing respective the edges (9,1), (9, 2), (9,3), (9, 4), (9,6) and (9, 7) with (10,1), (10, 2), (10,3), (10, 4), (10,6) and (10, 7). We get the final broadcast tree in which the transmission nodes are node 10 and node 8, see Figure 13.

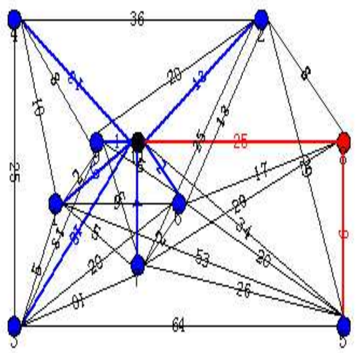


FIG. 13 – Final tree obtained by algorithm MBT–e_{MBT}=34

Finally, we set the transmission power of each transmission node and get the final minimum energy broadcast tree.

The transmission nodes are node 10 and node 8, the transmission power are e_{10}=25 and e_8=9. The algorithm Mini_Broad_Tree terminate. The total transmission power of the obtained minimum energy broadcast tree is

$$e_{MBT}=e_{10}+e_8=25+9=34,$$

and the receiving node is :

$$\beta_{MBT}=\beta_{10}+\beta_8=8+2=10$$

Conclusion

All of the three algorithms above only consider the total transmission power consumption in constructing the minimum-energy broadcast tree, but according to the discussion in the section **Operation Model**, we know that the receive power consumption is also very huge compare with the transmission power consumption. So in order to minimize effectively the energy consumption, we should minimize both the transmission and receive power consumption in constructing the minimum-energy broadcast tree. We named this problem is minimum receiving node and minimum energy broadcast tree problem.

In the following sections, we will introduced our algorithms which can efficient solve the minimum receiving node and minimum energy broadcast tree problem. We first adapt our conception in unicast environment, and prove that it is correct. Then we apply this conception to the multicast and broadcast environment to construct the minimum receiving node and minimum energy broadcast tree.

2 SPT IN ALL-WIRELESS NETWORKS

In a shortest-path problem, we are given a weighted, directed graph G(V, E), with weight function $\omega = E \rightarrow \Re$ mapping edges to real valued weights. The weight of a path $p = \{ v_0, v_1, ..., v_k \}$ is the sum of the weights of its constituent edges : $\omega(p) = \sum_{i=1}^{k} \omega(v_{i-1}, v_i)$. So we define the minimum shortest path weight from u to v :

$$\delta(u,v) = \begin{cases} \min\{\omega(p) : u \rightarrow^p v\} & \text{if there is a path p from u to v} \\ \infty & \text{otherwise} \end{cases}$$

In a Minimum Receiving Node Shortest Path Tree problem, for each edge(u, v) with the weight $\omega(u, v)$, we have the attribute $\beta(u, v)$ as the number of receiving nodes fall in transmission range of node u with transmission power $\omega(u, v)$. So, we have another two attributes : the number of receiving nodes of a path p is the sum of the receiving nodes of its constituent edges : $\beta(p) = \sum_{i=1}^{k} \beta(v_{i-1}, v_i)$. And the number of receiving node of MRN-SPT from u to v :

$$\gamma(u,v) = \begin{cases} \min\{\beta(p) : u \rightarrow^p v\} & \text{if there is a path p from u to v with} \delta(u,v) \\ \infty & \text{otherwise} \end{cases}$$

A shortest path from vertex u to vertexv is then defined as any path p with weight $\omega(p) = \delta(s,v)$, and a minimum receiving node shortest path from vertex u to vertexv is then defined as any path p with weight $\omega(p) = \delta(s,v)$ and with number of receiving node $\beta(p) = \gamma(s,v)$.

The algorithm **Dijkstra** was discovered by the pioneering mathematician and programmer E.W.Dijkstra. The algorithm is well known for single-source shortest paths tree construction. We will take a 9 nodes network example to introduce how to get a shortest path tree with this algorithm, see Figure 14.

The algorithm Dijkstra use the technique of *relaxation*. For each vertex $v \in V$, we maintain an attribute $d_{[v]}$, which is an upper bound on the weight of a shortest path from source node to node v. The process of relaxing an edge(u,v) consists of testing whether we can improve the shortest path to v found so far by going through u and, if so, updating $d_{[v]}$. A relaxation step may decrease the $d_{[v]}$ value.

We want to get a shortest path tree from the source node S to all other 8 nodes. **Initially**, the tree only constant source node, we focus on the source node and begin to determine the node which the source node can reach with the shortest path. We define the variable d_i(i=A, B, C, D, E, F, G, H) as the total weight of path from the source node to the node i. For node S

$d_A = \omega(S \rightarrow A) = \omega(S,\ A) = 9$ $\quad$ $d_E = \omega(S \rightarrow E) = \omega(S,\ E) = \infty$
$d_B = \omega(S \rightarrow B) = \omega(S,\ B) = 13$ $\quad$ $d_F = \omega(S \rightarrow F) = \omega(S,\ F) = \infty$
$d_C = \omega(S \rightarrow C) = \omega(S,\ C) = 25$ $\quad$ $d_G = \omega(S \rightarrow G) = \omega(S,\ G) = \infty$
$d_D = \omega(S \rightarrow D) = \omega(S,\ D) = 40$ $\quad$ $d_H = \omega(S \rightarrow H) = \omega(S,\ H) = \infty$

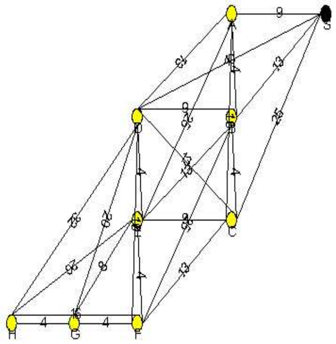


FIG. 14 – The example for algorithm Dijkstra SPT

So, the shortest path is from source node S to node A. We add node A to the tree. Thus, two nodes are included in the tree, see Figure 15.

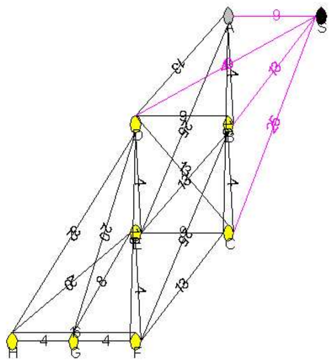


FIG. 15 – The example for algorithm Dijkstra step1 : S→A

Second, we focus on node A, to determine whether there are the paths through node A to the destinations have the less path weight and replace them. We can get :

$\omega(S \to A \to B) = \omega(S, A) + \omega(A, B) = 13 = \omega(S \to B)$
$\omega(S \to A \to C) = \omega(S, A) + \omega(A, C) = 25 = \omega(S \to C)$
$\omega(S \to A \to D) = \omega(S, A) + \omega(A, D) = 22 < \omega(S \to D)$
$\omega(S \to A \to E) = \omega(S, A) + \omega(A, E) = 34 < \omega(S \to E)$

So we have :

$d_A = \omega(S \to A) = 9$	$d_E = \omega(S \to A \to E) = 34$
$d_B = \omega(S \to B) = 13$	$d_F = \omega(S \to F) = \infty$
$d_C = \omega(S \to C) = 25$	$d_G = \omega(S \to G) = \infty$
$d_D = \omega(S \to A \to D) = 22$	$d_H = \omega(S \to H) = \infty$

Among the uncovered nodes, the shortest path is from source node S to node B. We add node B to the tree. Thus, three nodes are included in the tree, see Figure 16.

Third, we focus on node B, to determine whether there are the paths through node B to the destinations have the less path weight and replace them. We can get :

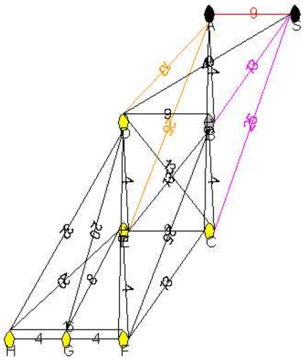


FIG. 16 – The example for algorithm Dijkstra step2 : S→B

$\omega(S \to B \to C) = \omega(S, B) + \omega(B, C) = 17 < \omega(S \to C)$
$\omega(S \to B \to E) = \omega(S, B) + \omega(B, E) = 26 < \omega(S \to A \to E)$
$\omega(S \to B \to F) = \omega(S, B) + \omega(B, F) = 38 < \omega(S \to F)$

So we have :

$d_A = \omega(S \to A) = 9$	$d_E = \omega(S \to B \to E) = 26$
$d_B = \omega(S \to B) = 13$	$d_F = \omega(S \to B \to F) = 38$
$d_C = \omega(S \to B \to C) = 17$	$d_G = \omega(S \to G) = \infty$
$d_D = \omega(S \to A \to D) = 22$	$d_H = \omega(S \to H) = \infty$

Among the uncovered nodes, the shortest path is from source node S through node B to node C. We add node C to the tree. Thus, four nodes are included in the tree, see Figure 17.

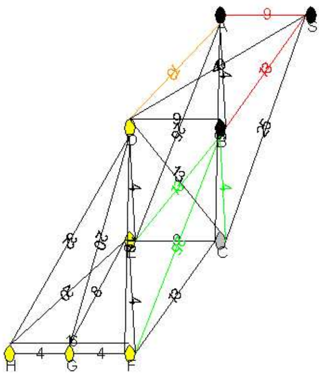
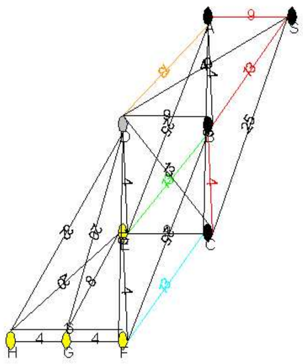


FIG. 17 – The example for algorithm Dijkstra step3 : B→C

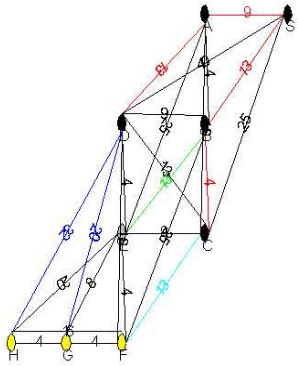
Continue,this procedure is continued until all nodes are included in the tree. as shown in Figure 18. The algorithm stop.

Finally, we get the final shortest path tree, see Figure 19. We can get the path weights from the source node S to all the destination nodes :

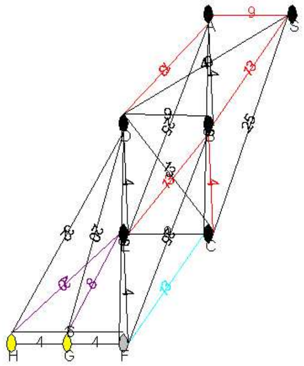
$d_A = \omega(S \to A) = 9$	$d_E = \omega(S \to B \to E) = 26$
$d_B = \omega(S \to B) = 13$	$d_F = \omega(S \to C \to F) = 30$
$d_C = \omega(S \to B \to C) = 17$	$d_G = \omega(S \to B \to E \to G) = 34$
$d_D = \omega(S \to A \to D) = 22$	$d_H = \omega(S \to B \to E \to G \to H) = 38$



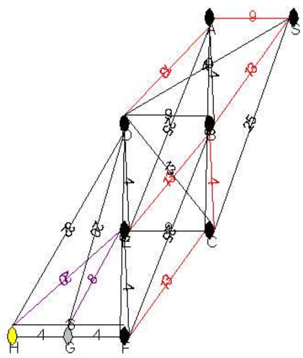
(a) Step 4 : A→D



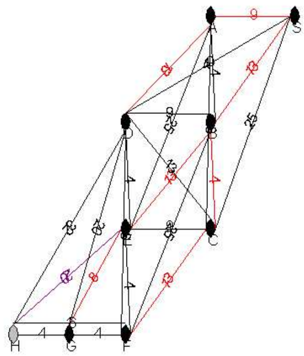
(b) Step 5 : B→E



(c) Step 6 : C→F



(d) Step 7 : E→G



(e) Step 8 : G→H

FIG. 18 – The example for algorithm SPT-step4-8

According to the final tree, we define the variable $n_i(i{=}A,\ B,\ C,\ D,\ E,\ F,\ G,\ H)$ as the total receiving node involved in the path from the source node to the node i. we can get the number of receiving node for each path above :

$n_A = \beta(S \to A) = 1$

$n_B = \beta(S \to B) = 2$

$n_C = \beta(S \to B) + \beta(B \to C) = 4$

$n_D = \beta(S \to A) + \beta(A \to D) = 4$

$n_E = \beta(S \to B) + \beta(B \to E) = 7$

$n_F = \beta(S \to C) + \beta(C \to F) = 8$

$n_G = \beta(S \to B) + \beta(B \to E) + \beta(E \to G) = 10$

$n_H = \beta(S \to B) + \beta(B \to E) + \beta(E \to G) + \beta(G \to H) = 12$

With the algorithm Dijkstra, we can get the good solution of the problem of shortest path tree. This algorithm only consider the least path weight(transmission power) which is very suitable in the wired networks, But in the *all-wireless* networks, a very famous property is the WMA[17], which will consider both the transmission and receiving energy in the transmission. So we have develop a new algorithm to construct a shortest path tree with the minimum receiving node along the paths. In the following subsection, we will give our conception of minimum receiving node shortest path tree which solve this problem very well.

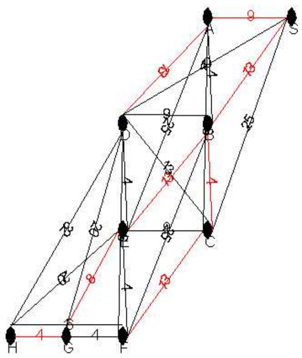


FIG. 19 – Shortest Path Tree obtained by algorithm Dijkstra

2.1 Presentation

In *all-wireless* network, a very useful and famous property is *WMA*[17], base on this property, we consider the receiving nodes in one node's transmission. Combined the shortest path problem described before, we then get the problem to find a minimum receiving node shortest path tree in the *all-wireless* networks. Given a weighted, directed graph G(V, E), with weight function $\omega = E \rightarrow \Re$ mapping edges to real valued weights , we want to establish a minimum receiving node shortest path tree : from any vertex u to others nodes in the graph G(V, E) with minimum cost and minimum receiving nodes along the paths.

In our conception, we also use the technology *relaxation*, but we do some adaptions. For each vertex $v \in V$, we maintain two attributes : one is $d_{[v]}$, which is an upper bound on the weight of a shortest path from source node to node v, and the other attribute $n_{[v]}$, which is an upper bound on the number of receiving node of a shortest path from source node to node v.

The process of relaxing an edge(u,v) consists of testing whether we can improve the shortest path to v found so far by going through u, if so, updating $d_{[v]}$ and $n_{[v]}$. If we get the same path power shortest paths to v find so far by going through u , testing whether we can improve the number of receiving node of the shortest path to v found so far by going through u. If so, updating $d_{[v]}$ and $n_{[v]}$. A relaxation step may decrease the $d_{[v]}$ and $n_{[v]}$ value.

We will take the same 9 nodes network example(see Figure 14) of SPT to introduce how to get the MRN-SPT. We act as the same procedure of SPT but we maintain two attributes in the process. **Initial**, for source node S, node A and node B, the same result of algorithm Dijkstra, see Figures 15, 16, 17 :

$$\begin{array}{llll}
d_A = \omega(S \rightarrow A) = 9 & n_A = 1 & d_E = \omega(S \rightarrow B \rightarrow E) = 26 & n_E = 7 \\
d_B = \omega(S \rightarrow B) = 13 & n_B = 2 & d_F = \omega(S \rightarrow B \rightarrow F) = 38 & n_F = 8 \\
d_C = \omega(S \rightarrow B \rightarrow C) = 17 & n_C = 4 & d_G = \omega(S \rightarrow G) = \infty & n_G = \infty \\
d_D = \omega(S \rightarrow A \rightarrow D) = 22 & n_D = 4 & d_H = \omega(S \rightarrow H) = \infty & n_H = \infty
\end{array}$$

Second, among the uncovered nodes, select the node with a minimum receiving node shortest path from the source node through the already covered nodes. We first determine the node with the shortest path, then if we get more than one nodes have the same shortest path weight, we compare the number of the receiving nodes along each paths and select the minimum one. Like this way, we can get the node with a minimum receiving node shortest path. Add the node selected to the final tree.

In the example, among the uncovered nodes, the minimum receiving node shortest path is from source node S through node B to node C. We add node C to the tree, for relaxation, we can get :

$$\omega(S \to B \to C \to E) = 26 = d_E \quad \beta(S \to B \to C \to E) = 6 < n_E = 7$$
$$\omega(S \to B \to C \to F) = 30 < d_F \quad \beta(S \to B \to C \to F) = 8$$

where $\omega(S \to B \to C \to E)$ is the path weight of path $S \to B \to C \to E$, $\beta(S \to B \to C \to E)$ is the number of receiving node involved in the path $S \to B \to C \to E$; $\omega(S \to B \to C \to F)$ is the path weight of path $S \to B \to C \to F$, $\beta(S \to B \to C \to F)$ is the number of receiving node involved in the path $S \to B \to C \to F$.

Notice : From above calculation, the path $S \to B \to C \to E$ has the same path weight with the path $S \to B \to E$, but less receiving node ; the path $S \to B \to C \to F$ has the less path weight than path $S \to B \to F$. We do path replace and update the attributes d_E, n_E and d_F, n_F(see Figure 20), we can get :

$d_A = \omega(S \to A) = 9 \quad n_A = 1 \quad d_E = \omega(S \to B \to C \to E) = 26 \quad n_E = 6$
$d_B = \omega(S \to B) = 13 \quad n_B = 2 \quad d_F = \omega(S \to B \to C \to F) = 30 \quad n_F = 8$
$d_C = \omega(S \to B \to C) = 17 \quad n_C = 4 \quad d_G = \omega(S \to G) = \infty \quad n_G = \infty$
$d_D = \omega(S \to A \to D) = 22 \quad n_D = 4 \quad d_H = \omega(S \to H) = \infty \quad n_H = \infty$

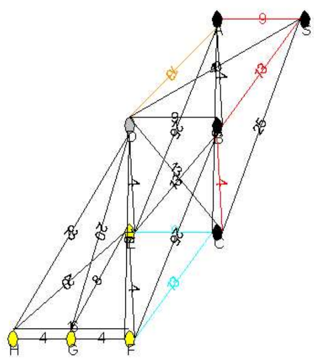


FIG. 20 – The example for algorithm MRN-SPT step4 : A→D

Among the uncovered nodes, nodes D has the shortest path from source node S, we add node D to the tree. Do relaxation for it, we can get :

$$\omega(S \to A \to D \to E) = 26 = d_E \quad \beta(S \to A \to D \to E) = 5 < n_E = 6$$
$$\omega(S \to A \to D \to G) = 42 < d_G \quad \beta(S \to A \to D \to G) = 10$$
$$\omega(S \to A \to D \to H) = 54 < d_H \quad \beta(S \to A \to D \to H) = 11$$

The path $S \to A \to D \to E$ has the same path weight but less receiving node than the path $S \to B \to C \to E$; the paths $S \to A \to D \to G$ and $S \to A \to D \to H$ have the less path weights than the origin. We do path replace and update d_E, d_G, d_H and n_E, n_G, n_H(see Figure 21) :

$d_A = \omega(S \to A) = 9 \quad n_A = 1 \quad d_E = \omega(S \to A \to D \to E) = 26 \quad n_E = 5$
$d_B = \omega(S \to B) = 13 \quad n_B = 2 \quad d_F = \omega(S \to B \to C \to F) = 30 \quad n_F = 8$
$d_C = \omega(S \to B \to C) = 17 \quad n_C = 4 \quad d_G = \omega(S \to A \to D \to G) = 42 \quad n_G = 10$
$d_D = \omega(S \to A \to D) = 22 \quad n_D = 4 \quad d_H = \omega(S \to A \to D \to H) = 54 \quad n_H = 11$

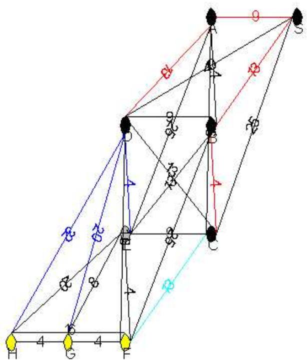
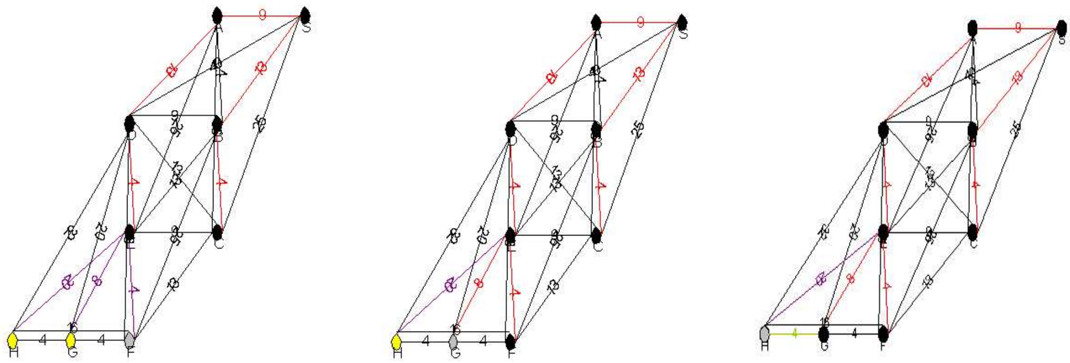


FIG. 21 – The example for algorithm MRN-SPT step5 : D→E

Continue, we repeat the procedure until all the nodes are included in the tree, as shown in Figure 22. The algorithm terminate.



(a) Step 6 : E→F (b) Step 7 : E→G (c) Step 8 : G→H

FIG. 22 – The example for algorithm MRN-SPT step6-8

Finally, we get the final minimum receiving node shortest path tree, see Figure 23. We can get the path weights and the number of receiving node from the source node S to all the destination nodes :

$d_A = \omega(S \to A) = 9$ $n_A = 1$
$d_B = \omega(S \to B) = 13$ $n_B = 2$
$d_C = \omega(S \to B \to C) = 17$ $n_C = 4$
$d_D = \omega(S \to A \to D) = 22$ $n_D = 4$
$d_E = \omega(S \to A \to D \to E) = 26$ $n_E = 5$
$d_F = \omega(S \to A \to D \to E \to F) = 30$ $n_F = 7$
$d_G = \omega(S \to A \to D \to E \to G) = 34$ $n_G = 8$
$d_H = \omega(S \to A \to D \to E \to G \to H) = 38$ $n_H = 10$

We have shown our conception and get the result by applying it to a network example, we then afford the official pseuco code and try to prove it in the next subsections.

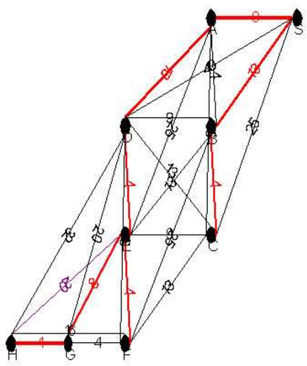


FIG. 23 – MRN Shortest Path Tree obtained by algorithm MRN-SPT

2.2 Pseuco Code of Algorithm MRN-SPT

Here we give the Pseuco code of algorithm Minimum Receiving Node Shortest Path Tree :
Definition :
- s : source node
- A : an n elements list
- $Adj[u]$: set of the neighbors of node u
- $d[v]$: weight of shortest path from node s to node v
- $\pi[v]$: parents node of node v
- $n[v]$: number of receiving nodes of shortest path from node s to node v
- $long[A]$: number of the elements in the list A
- $size[A]$: number of the arranged elements in the list A
- *left-child*(i) : the most left child node of node i
- *right-child*(i) : right child node of node i
- $\omega(u,v)$: the lowest weight from node u to node v
- $\beta(u,v)$: number of receiving node from node u to node v

Minimum_Receiving_Node_Shortest_Path_Tree_Algorithm( )
{
Initialize_ Single_ Source(G, s)
$S \leftarrow \emptyset$
$F \leftarrow \emptyset$
while $F \neq \emptyset$
  do $u \leftarrow$ ***Extract_ Min(F)***
    $S \leftarrow S \cup \{ u \}$
    for each node $v \in Adj[u]$
      do ***Relax(u,v,***ω***)***
}

Initialize_Single_Source(G, s)
{
for each node $v \in V$
  do $d[v] \leftarrow \infty$
    $\pi[v] \leftarrow 0$

```
          n[v] ← ∞
 d[v] ← 0
}
```

Relax(u,v,ω)

```
{
 for each node v' ∈ Adj[u], v' ≠ v
       do Node_In_Range(u,v,v')
 if d[v] > d[u]+ ω(u,v)
       then d[v] ← d[u]+ ω(u,v)
            π[v] ← u
            n[v] ← n[u]+ β(u,v)
 if d[v] = d[u]+ ω(u,v) and n[v] > n[u]+ β(u,v)
       then d[v] ← d[u]+ ω(u,v)
            π[v] ← u
            n[v] ← n[u]+ β(u,v)
}
```

Node_In_Range(u,v, v')

```
{
 β(u,v = 0
 if ω(u,v) > ω(u,v') then β(u,v) ← β(u,v) + 1
 return β(u,v)
}
```

Extract_Min(A)

```
{
 Build_Heap(A)
 if size[A] < 1
       then error "overflow negative"
 min ← A[1]
 A[1] ← A[size[A]]
 size[A] ← size[A]-1
 Heap(A,1)
 return min
}
```

Build_Heap(A)

```
{
 size[A] ← long[A]
 for i ← ⌊long[A]/2⌋ to 1
       do Heap(A,i)
}
```

```
Heap(A, i)
{
 l ← left-child(i)
 r ← right-child(i)
 if l ≤ size[A] and A[l] < A[i]
       then min ← l
       else min ← i
 if r ≤ size[A] and A[r] < A[min]
       then min ← r
 if min ≠ i
       then A[i] ← A[min]
       Heap(A,min
}
```

Now we got the pseuco code, and we have to prove that it is correct. The next subsection, we give the proof.

2.3 Proof

We prove the MRN-SPT algorithm : If we execute the algorithm of MRN-SPT on a weighted, directed graph G(V, E), with weight function $\omega = E \rightarrow \Re$ with the source $s \in V$, then after the execution, we have $d[v]$ $=\delta(s,v)$, $n[v]$ $=\gamma(s,v)$ for all of the nodes $v \in V$

S : the set of vertices for which the shortest paths (from s) have been obtained already.

F : priority queue, each vertex in V is placed in it (In general, F will hold each vertex in V-S)

Base : Initially $S = \emptyset$ and hence the invariant is trivially true.

Inductive Step : Assume that when $|S| = i$, the invariant holds (i.e. $\forall\ u \in S$, $d[u]$ $=\delta(s,u)$, $n[u]$ $=\gamma(s,u)$). We now prove that the invariant will be true when $|S| = i + 1$

Let v be the $(i+1)^{st}$ vertex extracted from F (and hence placed in S) and let p be a path from s to v with weight $d[v] = \omega(p)$ and nodes in the transmission range $n[v] = \beta(p)$. Let u be the vertex just before v in path p. Since paths to vertices in F are only considered that use vertices from S, $u \in S$ and hence by the inductive hypothesis $d[u]$ $=\delta(s,u)$, $n[u]$ $=\gamma(s,u)$).

We first prove that p is a shortest path from s to v, that is $d[v] = \omega(p) = \delta(s,v)$. Assume the contrary, the node v is the first node which $d[v] \neq \delta(s,v)$, (i.e. there exists a path p^* from s to v, where $\omega(p^*) < \omega(p)$. Since p^* connects a vertex $s \in S$ to a vertex $v \in V$-S, consider there is a first node b along path p^* with $b \in V$-S, and get node $a \in S$ just before b as the predecessor of b. So there must be a first edge $(a,\ b) \in p^*$ where $a \in S$ and $b \in V$-S. We can partition the path p^* as $s \rightarrow^{p_1} a \rightarrow b \rightarrow^{p_2} v$, where the portion of p^* from s to a is named path p_1 and the portion of p^* from b to v is named path p_2, see Figure 24. By the inductive hypothesis, we have $d[a]$ $=\delta(s,a)$, $n[a]$ $=\gamma(s,a)$). Furthermore since p^* is a shortest path, according to the lemma 1 [5], it follows that $s \rightarrow a \rightarrow^{p_1} b$ must be a shortest path from s to b. When a was placed in S, the edge $(a,\ b)$ was considered and according to lemma 6 [6], hence $d[b]$ $=\delta(s,b)$. Since b appear before v along the minimum receiving nodes shortest path p^* from s to v and the edge weights are all nonnegative, we can get $\delta(s,b) < \delta(s,v)$, so :

[5] see following, we give the description and proof

[6] see following, we give the description and proof

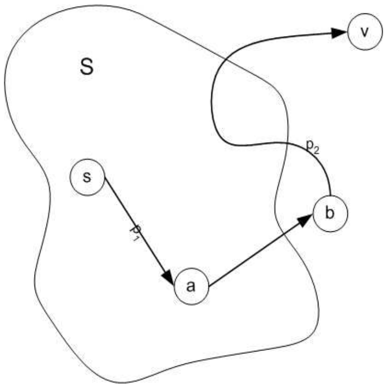


FIG. 24 – Proof for MRN-SPT

$$\begin{aligned} d[b] &= \delta(s,b) \\ &\le \delta(s,v) = \omega\ (p^*) \\ &\le d[v] \end{aligned}$$

But since v was the $(i+1)^{st}$ vertex from F while b was still in F, it follows that $d[v] = \omega\ (p) \le d[b]$. So we can get $d[b] = \delta(s,b) = \delta(s,v) = d[v]$ which contradicts that $d[v] \ne \delta(s,v)$. Completing the inductive proof.

Now we prove that p is a shortest path involved minimum receiving nodes in the transmission range from s to v, that is $d[v] = \omega(p) = \delta(s,v)$ and $n[v] = \beta(p) = \gamma(s,v)$. Assume the contrary, the node v is the first node which $d[v] = \delta(s,v)$, $n[v] \ne \gamma(s,v)$ (i.e. there exists a path p^* from s to v, where $\omega(p^*) = \omega(p)$, but $\beta(p^*) = \beta(p)$. Since p^* connects a vertex $s \in S$ to a vertex $v \in V$-S, consider there is a first node b along path p^* with $b \in V$-S, and get node $a \in S$ just before b as the predecessor of b. So there must be a first edge $(a,\ b) \in p^*$ where $a \in S$ and $b \in V$-S. We can partition the path p^* as $s \rightarrow^{p_1} a \rightarrow b \rightarrow^{p_2} v$, where the portion of p^* from s to a is named path p_1 and the portion of p^* from b to v is named path p_2, see Figure 22. By the inductive hypothesis, we have $d[a] =\delta(s,a)$, $n[a] =\gamma(s,a))$. Furthermore since p^* is a shortest path with minimum receiving nodes, according to the lemma 1, it follows that $s \rightarrow^{p_1} a \rightarrow b$ must be a shortest path with minimum receiving nodes from s to b. When a was placed in S, the edge $(a,\ b)$ was considered and according to lemma 6, hence $d[b] =\delta(s,b)$, $n[b] =\gamma(s,b)$. Since b appear before v along the minimum receiving nodes shortest path p^* from s to v, the edge weights and the number of receiving nodes are all nonnegative, we can get $\delta(s,b) \le \delta(s,v)$ and $\gamma(s,b) \le \gamma(s,v)$, so :

$$\begin{aligned} d[b] &= \delta(s,b) \\ &\le \delta(s,v) = \omega(p^*) \\ &\le d[v] \\ n[b] &= \gamma(s,b) \\ &\le \gamma(s,v) = \beta(p^*) \\ &\le n[v] \end{aligned}$$

But since v was the $(i+1)^{st}$ vertex from F while b was still in F, it follows that $d[v] = \omega\ (p) \le d[b]$ or $d[v] = \omega\ (p) = d[b]$ but with $n[v] = \beta\ (p) \le n[b]$. So we can get $d[b] = \delta(s,b) = \delta(s,v) = d[v]$ and $n[b] = \gamma(s,b) = \gamma(s,v) = n[v]$ which contradicts that $d[v] \ne \delta(s,v)$, $n[v] \ne \gamma(s,v)$. Completing the inductive proof.

We have finished the proof for our algorithm, but our algorithm is based on two lemmas(Lemma 1 and Lemma 6). We next give the proof of these two Lemmas.

Lemma 1 : The sub-paths of the minimum receiving node shortest path are also the minimum receiving node shortest path.

Given a weighted, directed graph G(V, E), $\omega = E \rightarrow \Re$, assume that $p = \{ v_1, v_2, ..., v_k \}$ is a minimum receiving node shortest path from a node v_1 to a node v_k, and for any pair integer i,j with $1 \leq i \leq j \leq k$, assume $p_{ij} = \{ v_i, v_{i+1}, ..., v_j \}$ is a sub-path of p which is from node v_i to node v_j. Then, p_{ij} is the minimum receiving node shortest path from node v_i to node v_j.

Demonstration : We first prove that the sub path is the shortest path. If we separate the path p into $v_1 \rightarrow^{p_{1i}} v_i \rightarrow^{p_{ij}} v_j \rightarrow^{p_{jk}} v_k$, p_{1i} is the path from v_1 to v_i ; p_{ij} is the path from v_i to v_j ; p_{jk} is the path from v_j to v_k, then we have $\omega(p) = \omega(p_{1i}) + \omega(p_{ij}) + \omega(p_{jk})$ with $\beta(p) = \beta(p_{1i}) + \beta(p_{ij}) + \beta(p_{jk})$. Assume that there is a path $p_{ij}^{'}$ from v_i to v_j with $\omega(p_{ij}^{'}) < \omega(p_{ij})$. In this case, $v_1 \rightarrow^{p_{1i}} v_i \rightarrow^{p_{ij}^{'}} v_j \rightarrow^{p_{jk}} v_k$ is the path from v_1 to v_k with the weight $\omega(p_{1i}) + \omega(p_{ij}^{'}) + \omega(p_{jk})$ lower to $\omega(p)$, that is $\omega(p_{1i}) + \omega(p_{ij}^{'}) + \omega(p_{jk}) < \omega(p)$, this contradicts our assumption that p is the minimum receiving node shortest path from v_1 to v_k.

Now we prove that the sub path also involved minimum receiving node. Assume that there is another path p_{ij}^{*} from v_i to v_j with $\omega(p_{ij}^{*}) = \omega(p_{ij})$ but $\beta(p_{ij}^{*}) < \beta(p_{ij})$. In this case, $v_1 \rightarrow^{p_{1i}} v_i \rightarrow^{p_{ij}^{*}} v_j \rightarrow^{p_{jk}} v_k$ is the path from v_1 to v_k with the weight $\omega(p_{1i}) + \omega(p_{ij}^{*}) + \omega(p_{jk}) = \omega(p)$ but with the receiving nodes $\beta(p_{1i}) + \beta(p_{ij}^{*}) + \beta(p_{jk}) < \beta(p)$, this contradicts our assumption that p is the minimum receiving node shortest path from v_1 to v_k.

Corollary 2 :Given a weighted, directed graph G(V, E), $\omega = E \rightarrow \Re$, Assume that a MRN-SPT p from the origin s to node v can be divide as $s \rightarrow^{p'} u \rightarrow v$, u is a node and p' is a path. Then the weight of the MRN-SPT p from s to v equal to $\delta(s,v) = \delta(s,u) + \omega(u,v)$, the number of receiving node of the MRN-SPT p from s to v equal to $\gamma(s,v) = \gamma(s,u) + \beta(u,v)$.

Demonstration : From the lemma 1, the sub-path p' is also the MRN-SPT from s to u. So,

$$\begin{aligned}
\delta(s,v) &= \omega(p) \\
&= \omega(p') + \omega(u, v) \\
&= \delta(s,u) + \omega(u, v) \\
\gamma(s,v) &= \beta(p) \\
&= \beta(p') + \beta(u, v) \\
&= \gamma(s,u) + \beta(u, v)
\end{aligned}$$

Lemma 3 :Given a weighted, directed graph G(V, E), $\omega = E \rightarrow \Re$, a source node s. Then for all the edges$(u,v) \in E$, we get $\delta(s,v) \leq \delta(s,u) + \omega(u, v)$ and specially for $\delta(s,v) = \delta(s,u) + \omega(u, v)$, we can also get $\gamma(s,v) \leq \gamma(s,u) + \beta(u, v)$.

Demonstration : A MRN-SPT p from s to v has a weight and receiving nodes lower or equal to any other paths from s to v. In particular, a path p has a weight and receiving nodes lower or equal to a path constitute by a MRN-SPT from s to u plus the edge(u,v).

Lemma 4 :Given a weighted, directed graph G(V, E), $\omega = E \rightarrow \Re$, and $(u,v) \in E$. Then, immediately after the relax of the edge(u,v) by calling **Relax(u,v,ω)**, we get $d[v] \leq d[u] + \omega(u,v)$, and specially for $d[v] = d[u] + \omega(u,v)$, we can also get $n[v] \leq n[u] + \beta(u,v)$.

Demonstration : If just before relax an edge(u,v), we have $d[v] > d[u] + \omega(u,v)$, then after the relax $d[v] = d[u] + \omega(u,v)$ and $n[v] = n[u] + \beta(u,v)$;

If before the relax we have $d[v] = d[u] + \omega(u,v)$ and $n[v] > n[u] + \beta(u,v)$, then $d[v] = d[u] + \omega(u,v)$ and $n[v] = n[u] + \beta(u,v)$ after the relax ;

If before the relax we have $d[v] = d[u] + \omega(u,v)$ and $n[v] \leq n[u] + \beta(u,v)$, then after the relax neither $d[v]$ nor $n[v]$ will be modified.

If before the relax we have $d[v] < d[u] + \omega(u,v)$, then after the relax neither $d[v]$ nor $n[v]$ will be modified.

Lemma 5 :Given a weighted, directed graph G(V, E), ω = E $\rightarrow \Re$, node $s \in V$ is the source. We assume that the graph is initialised by **Initialize_Single_Source(G, s)**. Then, $d[v] \geq \delta(s,\ v)$ with $n[v] \geq \gamma(s,\ v)$ for $v \in V$, and these invariants are preserved for any sequence of the steps of relax on the edges of G. Furthermore, once $d[v]$ reach $\delta(s,\ v)$ and $n[v]$ reach $\gamma(s,\ v)$, they will not be changed.

Demonstration : The invariant $d[v] \geq \delta(s,\ v)$ was verified after the initialization, since $d[s] = 0 \geq \delta(s,\ s)$ and $d[v] = \infty$ implicate $d[v] \geq \delta(s,\ v)$ for all $v \in V$-{ s } ; we will indicate that the invariants are preserved for any sequence of steps of relax. Assume v is the first node for which one step of relaxes of edge($u,\ v$) produced $d[v] < \delta(s,\ v)$. Then just after the relax of the edge($u,\ v$), we have :

$$
\begin{aligned}
d[u] + \omega(u,\ v) &= d[v] \\
&= < \delta(s,\ v) \\
&= \leq \delta(s,u) + \omega(u,\ v)
\end{aligned}
$$

This implicate $d[u] < \delta(s,\ u)$. But, $d[u]$ isn't modified by the relax of the edge($u,\ v$), this inequality must have been verified just before the relax of the edge, this contradict that v is the first node for which one step of relax of edge($u,\ v$) produced $d[v] < \delta(s,\ v)$. We can conclude that the invariant $d[v] \geq \delta(s,\ v)$ was preserved for all $v \in V$.

The invariant $n[v] \geq \gamma(s,\ v)$ was verified after the initialization, since $n[s] = 0 \geq \gamma(s,\ s)$ and $n[v] = \infty$ implicate $n[v] \geq \gamma(s,\ v)$ for all $v \in V$-{ s } ; we will indicate that the invariants are preserved for any sequence of steps of relax. Assume v is the first node for which one step of relaxes of edge($u,\ v$) produced $d[v] = \delta(s,\ v)$ but $n[v] < \gamma(s,\ v)$. Then just after the relax of the edge($u,\ v$), when $d[v] = d[u] + \omega(u,\ v)$ we have :

$$
\begin{aligned}
n[u] + \beta(u,\ v) &= n[v] \\
&= < \gamma(s,\ v) \\
&= \leq \gamma(s,u) + \beta(u,\ v)
\end{aligned}
$$

This implicate $n[u] < \gamma(s,\ u)$. But, $n[u]$ isn't modified by the relax of the edge($u,\ v$), this inequality must have been verified just before the relax of the edge, this contradict that v is the first node for which one step of relax of edge($u,\ v$) produced $n[v] < \gamma(s,\ v)$. We can conclude that the invariant $n[v] \geq \gamma(s,\ v)$ was preserved for all $v \in V$.

Lemma 6 :Given a weighted, directed graph G(V, E), ω = E $\rightarrow \Re$, node $s \in V$ is the source, and $s \rightarrow u \rightarrow v$ is the MRN-SPT for two given nodes $u,\ v \in V$. We assume that G is initialized by **Initialize_Single_Source(G, s)** and next execute a sequence of steps of the relax on the edges of G by calling **Relax(u,v, ω)**. If $d[u] = \delta(s,\ u)$ with $n[u] = \gamma(s,\ u)$ after the call.

Demonstration : If $d[u] = \delta(s,\ u)$ with $n[u] = \gamma(s,\ u)$ before the relax of edge($u,\ v$), then after the relax we can get : when

$$
\begin{aligned}
d[v] &< d[u] + \omega(u,\ v) \\
&= \delta(s,\ u) + \omega(u,\ v) \\
&= \delta(s,\ v)
\end{aligned}
$$

but according to the lemma 5, we can get that $d[v] \geq \delta(s,\ v)$, so we can conclude that $d[v] = \delta(s,\ v)$. Under this condition, the receiving node $n[v]$ will not be changed and will be $n[v] = \gamma(s,\ v)$;

when

$$
\begin{aligned}
d[v] &= d[u] + \omega(u,\ v) \\
&= \delta(s,\ u) + \omega(u,\ v) \\
&= \delta(s,\ v)
\end{aligned}
$$

we have

$$n[v] \leq n[u] + \beta(u,\ v)$$
$$= \gamma(s,\ u) + \beta(u,\ v)$$
$$= \gamma(s,\ v)$$

but according to the lemma 5, we can get that $d[v] \geq \delta(s,\ v)$, $n[v] \geq \gamma(s,\ v)$, so we can conclude that $d[v] = \delta(s,\ v)$, $n[v] = \gamma(s,\ v)$.

2.4 Result

In the above sections, we describe the shortest path tree problem and minimum receiving node shortest path tree problem, we also give the pseuco code of the algorithm minimum receiving node shortest path tree, and have proved its correction. Now we will compare the the path weight and the number of receiving node of the results of applying algorithm Dijkstra and MRN-SPT with the same network example.

	Dijkstra-SPT		MRN-SPT	
	d_i	n_i	d_i	n_i
$i{=}A$	9	1	9	1
$i{=}B$	13	2	13	2
$i{=}C$	17	4	17	4
$i{=}D$	22	4	22	4
$i{=}E$	26	7	26	5
$i{=}F$	30	8	30	7
$i{=}G$	34	10	34	8
$i{=}H$	38	12	38	10

In the example, we can get that all the paths of MRN-SPT have the same path weight with the paths of SPT, but almost all the paths have less receiving node than the ones of SPT. Because both in SPT and MRN-SPT, if the algorithms meet the new paths with less path weight than the origin, it will do same to update the path to the new one, but when the algorithm meet the new paths with the same path weight than the origin, the algorithm SPT do not do anything, but by contraries, under this condition, the algorithm MRN-SPT compare the number of receiving node involved in the two paths, and select path with the less receiving node to update. So after check all the nodes in the graph, the path weight should be same with two algorithms, but the algorithm MRN-SPT must have less receiving node involved in the tree than the one of algorithm SPT.

3 PROPOSED ALGORITHMS

After success in applying our conception on the uniticast environment in *all-wireless* networks, we want to apply it on the multicast and broadcast environment. We adapted the two algorithms we have described in the section **Relative Algorithms** with our conception to fit the requirement of minimizing transmission and receive power consumption in a broadcast tree in *all-wireless* networks. The original algorithms are algorithm *EWMA* and *BIP*, and after adopt our conception, we call them algorithm Minimum Receiving Node Embedded Wireless Multicast Advantage(MRN-EWMA) and algorithm Minimum Receiving Node Broadcast Incremental Power(MRN-BIP). In the following sections, we will describe these algorithms in details.

3.1 Minimum Receiving Node Embedded Wireless Multicast Advantage

This algorithm is base on the algorithm EWMA which is supported by Čagalj et al.[16]. The original algorithm begin with a link-based minimum spanning tree(MST), and improve it by exchanging some existing branches in the MST for new ones to decrease the total transmission power necessary to maintain the broadcast tree.

The algorithm MRN-EWMA begin with a feasible solution(an initial feasible broadcast tree–Minimum Receiving Node Minimum Spanning Tree) for a given network. Then it improve the solution by exchanging some existing branches in the initial tree for new branches to decrease the total energy necessary to maintain the broadcast tree, just the algorithm EWMA does. If among the processes of exchanging, there are more than one process of exchanging cause the same total energy decrease of the broadcast tree, we consider the decrease of the total of receiving node of the broadcast tree caused by every one of these processes. Do select the process of exchanging which has the maximum decrease of the total receiving node. Thus, after some repeat step of MRN-EWMA, the remain transmission nodes can additionally increase their transmission power to reach more nodes but decrease the number of transmission nodes to lower the total transmission power and the receiving nodes.

Firstly, we have to construct MRN-MST, it's difference with the MST is the standard of select node to add to the tree. The algorithm **Prim**'s algorithm add the new node only consider minimum link cost, if there are more than one edges have the same minimum link costs, it randomly select one to add to the tree. But the algorithm MRN-MST will consider both link cost and receiving node involved, so when the algorithm meet more than one edges have the same minimum link cost, it will compare their receiving node involved by their edges respectively and select the one with less receiving node to add to the tree.

Here, we will introduce how to construct a MRN-MST with a 10 node network example in which node 10 is the source node, see Figure 25. In our algorithm, for each vertex $v \in V$, we maintain four attributes : one is $C_{[v]}$, which is an upper bound on the weight of a link to node v, and the attribute $n_{[v]}$, which is an upper bound on the number of receiving node involved in a link to node v ; $e_{[v]}$ is the transmission power of node v ; $\beta_{[v]}$ is the number of receiving node involve in the transmission range of node v with transmission power $e_{[v]}$.

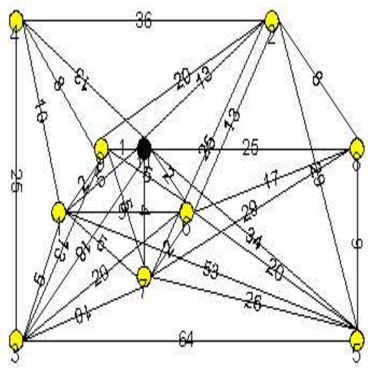


FIG. 25 – The example for algorithm MRN-MST

We want to get a minimum receiving node MST, **initially**, we want to find the node that the source node can reach with minimum link cost. In the example, we have :

$C_{[1]}=\omega(10 \rightarrow 1) = 5$ $n_{[1]}=\beta(10 \rightarrow 1) = 4$,
$C_{[2]}=\omega(10 \rightarrow 2) = 13$ $n_{[2]}=\beta(10 \rightarrow 2) = 6$,
$C_{[3]}=\omega(10 \rightarrow 3) = 18$ $n_{[3]}=\beta(10 \rightarrow 3) = 7$,
$C_{[4]}=\omega(10 \rightarrow 4) = 13$ $n_{[4]}=\beta(10 \rightarrow 4) = 6$,
$C_{[5]}=\omega(10 \rightarrow 5) = 34$ $n_{[5]}=\beta(10 \rightarrow 5) = 9$
$C_{[6]}=\omega(10 \rightarrow 6) = 2$ $n_{[6]}=\beta(10 \rightarrow 6) = 2$
$C_{[7]}=\omega(10 \rightarrow 7) = 4$ $n_{[7]}=\beta(10 \rightarrow 7) = 3$
$C_{[8]}=\omega(10 \rightarrow 8) = 25$ $n_{[8]}=\beta(10 \rightarrow 8) = 8$
$C_{[9]}=\omega(10 \rightarrow 9) = 1$ $n_{[9]}=\beta(10 \rightarrow 9) = 1$

Among all the nodes, only node 9 has the minimum link cost from node 10, we add node 9 in the tree, see Figure 26, we have :

$$e_{[10]}=\omega(10 \rightarrow 9) = 1 \quad \beta_{[10]}=\beta(10 \rightarrow 9) = 1$$

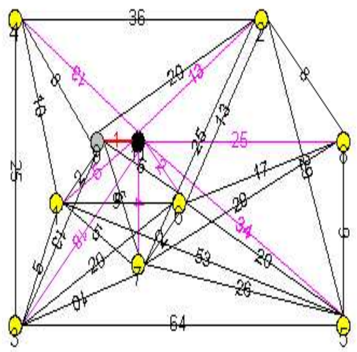


FIG. 26 – The example for algorithm MRN-MST step1 : 10→9

Second, we apply the algorithm on the newly added node, and get the nodes which the newly added node can reach with less edge weight, if necessary with less receiving node.

In the example, we get the nodes which node 9 can reach with less edge weight, calculate the edge weight and the number of receiving node involved in the edge. They are node 1, 3 and 4. We have :

$$\begin{array}{ll} \omega(9 \rightarrow 1) = 2 < C_{[1]}=5 & n_{[1]}=\beta(9 \rightarrow 1) = 2 \\ \omega(9 \rightarrow 3) = 13 < C_{[3]}=18 & n_{[3]}=\beta(9 \rightarrow 3) = 6 \\ \omega(9 \rightarrow 4) = 8 < C_{[4]}=13 & n_{[4]}=\beta(9 \rightarrow 4) = 5 \end{array}$$

So we update the attributes and get :

$$\begin{array}{llll} C_{[1]}=\omega(9 \rightarrow 1) = 2 & n_{[1]}=\beta(9 \rightarrow 1) = 2, & C_{[6]}=\omega(10 \rightarrow 6) = 2 & n_{[6]}=\beta(10 \rightarrow 6) = 2 \\ C_{[2]}=\omega(10 \rightarrow 2) = 13 & n_{[2]}=\beta(10 \rightarrow 2) = 6, & C_{[7]}=\omega(10 \rightarrow 7) = 4 & n_{[7]}=\beta(10 \rightarrow 7) = 3 \\ C_{[3]}=\omega(9 \rightarrow 3) = 13 & n_{[3]}=\beta(9 \rightarrow 3) = 6, & C_{[8]}=\omega(10 \rightarrow 8) = 25 & n_{[8]}=\beta(10 \rightarrow 8) = 8 \\ C_{[4]}=\omega(9 \rightarrow 4) = 8 & n_{[4]}=\beta(9 \rightarrow 4) = 5, & C_{[9]}=\omega(10 \rightarrow 9) = 1 & n_{[9]}=\beta(10 \rightarrow 9) = 1 \\ C_{[5]}=\omega(10 \rightarrow 5) = 34 & n_{[5]}=\beta(10 \rightarrow 5) = 9 & & \end{array}$$

Third, among all the uncovered nodes, select the edge(covered node→uncovered node) with minimum edge weight, and add the destination of this edge to the tree. If there are more than one edges have the same minimum edge weight, consider respectively the number of receiving node involved in the tree if these edges added, and select the one with minimum receiving node.

Among all the nodes, both the edge(10, 6) and edge(9, 1) have the minimum link cost, we compare the number of receiving node involved of the tree by adding node 6 or node 1 to the tree. If we add node 6 to the tree, the number of receiving node of the tree will be $n_{[6]}=\beta(10 \rightarrow 6) = 2$, and for node 1, the number of receiving node of the tree will be $n_{[9]}+n_{[1]}=\beta(10 \rightarrow 9)+\beta(9 \rightarrow 1) = 3$, We add node 6 to the tree, see Figure 27, we have :

$$e_{[10]}=\omega(10 \rightarrow 6) = 2 \quad \beta_{[10]}=\beta(10 \rightarrow 6) = 2$$

Continue, we repeat the last procedures until all the uncovered node are included in the tree.

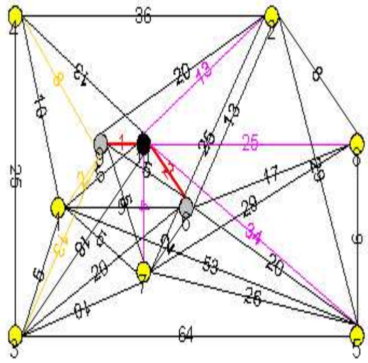


FIG. 27 – The example for algorithm MRN-MST step2 : 10→6

We apply the algorithm on the node 6, we get the nodes which node 6 can reach with less edge weight, calculate the edge weight and the number of receiving node involved in the edge. We have :

$$\omega(6 \to 5) = 20 < C_{[5]}{=}34 \quad n_{[5]}{=}\beta(6 \to 5) = 8$$
$$\omega(6 \to 7) = 2 < C_{[7]}{=}4 \quad n_{[7]}{=}\beta(6 \to 7) = 2$$
$$\omega(6 \to 8) = 17 < C_{[8]}{=}25 \quad n_{[8]}{=}\beta(6 \to 8) = 6$$

So we update the attributes and get :

$$C_{[1]}{=}\omega(9 \to 1) = 2 \quad n_{[1]}{=}\beta(9 \to 1) = 2 \qquad C_{[6]}{=}\omega(10 \to 6) = 2 \quad n_{[6]}{=}\beta(10 \to 6) = 2$$
$$C_{[2]}{=}\omega(10 \to 2) = 13 \quad n_{[2]}{=}\beta(10 \to 2) = 6 \qquad C_{[7]}{=}\omega(6 \to 7) = 2 \quad n_{[7]}{=}\beta(6 \to 7) = 2$$
$$C_{[3]}{=}\omega(9 \to 3) = 13 \quad n_{[3]}{=}\beta(9 \to 3) = 6 \qquad C_{[8]}{=}\omega(6 \to 8) = 17 \quad n_{[8]}{=}\beta(6 \to 8) = 6$$
$$C_{[4]}{=}\omega(9 \to 4) = 8 \quad n_{[4]}{=}\beta(9 \to 4) = 5 \qquad C_{[9]}{=}\omega(10 \to 9) = 1 \quad n_{[9]}{=}\beta(10 \to 9) = 1$$
$$C_{[5]}{=}\omega(6 \to 5) = 20 \quad n_{[5]}{=}\beta(6 \to 5) = 8$$

Among all the nodes, both the edge(6, 7) and edge(9, 1) have the minimum link cost, we compare the receiving node involved of add node 7 and node 1 to the tree. If we add node 7 to the tree, the number of receiving node will be $n_{[6]}+n_{[7]}=\beta(10 \to 6)+\beta(6 \to 7)= 4$. For node 1, the number of receiving node will be $n_{[6]}+n_{[1]}=\beta(10 \to 6)+\beta(9 \to 1) = 4$. We randomly add node 1 to the tree, see Figure 28, we have :

$$e_{[10]}{=}\omega(10 \to 6) = 2 \quad \beta_{[10]}{=}\beta(10 \to 6) = 2$$
$$e_{[9]}{=}\omega(9 \to 1) = 2 \quad \beta_{[9]}{=}\beta(9 \to 1) = 2$$

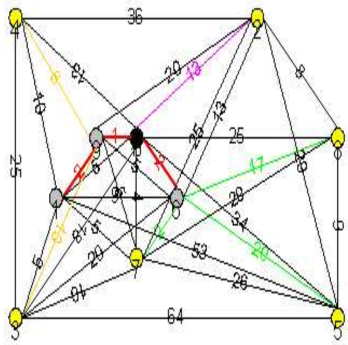
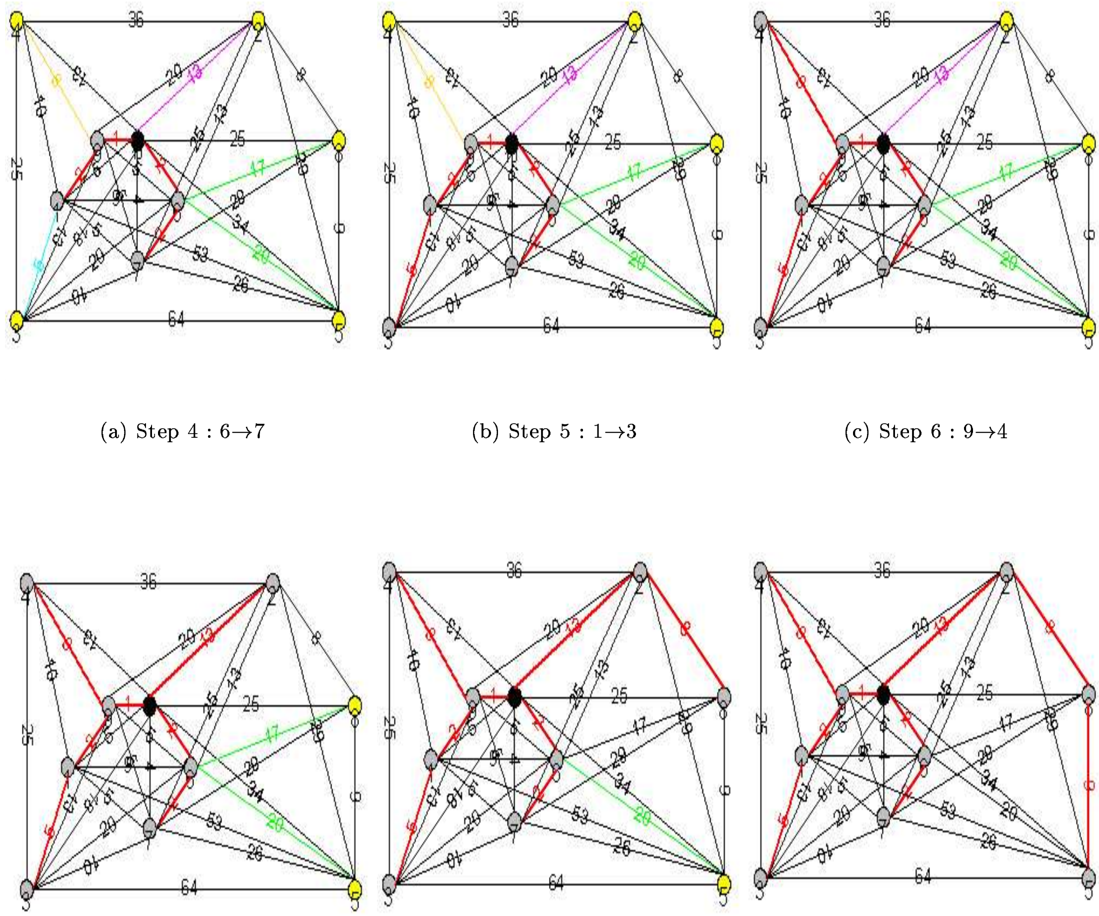


FIG. 28 – The example for algorithm MRN-MST step3 : 9→1

We continue this procedure until all nodes are included in the tree, as shown in Figure 29.



(a) Step 4 : 6→7 (b) Step 5 : 1→3 (c) Step 6 : 9→4

(d) Step 7 : 10→2 (e) Step 8 : 2→8 (f) Step 9 : 8→5

FIG. 29 – The example for algorithm MRN-MST step4-9

Finally, we have the final MRN-MST. In the example, the MRN-MST(Figure 30) in which the transmission nodes are nodes 10, 9, 1, 6, 2, 8.

Secondly, after getting the MRN-MST, we apply algorithm MRN-EWMA on it. MRN-EWMA determine the respective transmission power gain and receiving node gain of the nodes in the set C-F-E one by one to build a broadcast tree.

As the description of algorithm EWMA, we get the set C-F-E contains just the source node 10. So, we determine the maximum transmission power gain and receiving node gain of node 10 when exclude the other transmission nodes in the initial MST. Because the initial transmission power of node 10 is e_{10}=13, so nodes 1, 2, 4, 6, 7 and 9 are included in the transmission range of node 10. Exactly the node 6 and node 9 are no more the transmission nodes. Thus for example, there are only three transmission nodes we can try to exclude : node 1, 2 and 8.

See Figure 30, in order to exclude node 1 :

the source node 10 has to increase its transmission power :	$\Delta e_{10}^{1}=\omega_{10,3}$-$e_{10}$=18-13=5
the transmission power gain for exclude node 1 (g_{10}^{1}) :	$g_{10}^{1}=e_1$-Δe_{10}^{1}=5-5=0

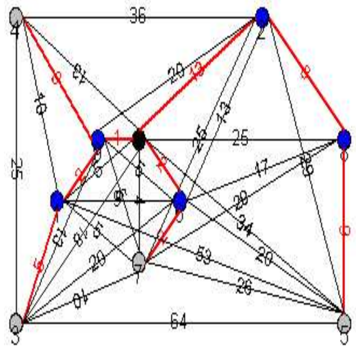


FIG. 30 – Final tree obtained by algorithm MRN-MST

where $\omega_{10,3}$ is the transmission power from source node 10 to node 3, e_1 is the energy at which node 1 transmits in MRN-MST.

In order to exclude node 2 :

the source node 10 has to increase its transmission power : Δe_{10}^2=$\omega_{10,8}$-e_{10}=25-13=12
the transmission power gain for excluding node 2 (g_{10}^2) : g_{10}^2=e_1+e_2-Δe_{10}^2=5+8-12=1

where $\omega_{10,8}$ is the transmission power from source node 10 to node 8, e_1 and e_2 is the respective energy at which node 1 and node 2 transmits in MRN-MST. With the transmission power of node 10 increase to e_{10}=25, the node 3 and 8 will fall in the transmission range of node 10, so the node 1 and 2 will not be transmission nodes.

In order to exclude node 8 :

the source node 10 has to increase its transmission power : Δe_{10}^8=$\omega_{10,5}$-e_{10}=34-13=21
the transmission power gain for excluding node 8 (g_{10}^8) : g_{10}^8=e_1+e_2+e_8-Δe_{10}^8=5+8+9-21=1

where $\omega_{10,5}$ is the transmission power from source node 10 to node 5, e_1, e_2 and e_8 are the respective energy at which node 1, node 2 and node 8 transmits in MRN-MST. With the transmission power of node 10 increase to e_{10}=34, the node 3, 5 and 8 will fall in the transmission range of node 10, so the node 1, 2 and 8 will not be transmission nodes.

Thirdly, have the gains for all nodes from C-F-E, the MRN-EWMA select a node with the highest positive gain in the set F. If there are more than one gain have the same highest positive gain, select the node with highest positive receiving node gain. If none has the positive gain, MRN-EWMA selects the node that includes its *child* nodes[7] in MRN-MST at minimum energy. After that, MRN-EWMA adds all the nodes that this node excludes to the set E.

In the example, node 10 is the only one in the set C-F-E, and excluding node 2 and node 8 have the same highest positive transmission power gain, so we have to calculate the receiving node gain : For excluding node 2 :

the source node 10 increase its number of receiving node : $\Delta\beta_{10}^2$=$\beta_{10,8}$-β_{10}=8-6=2
the receiving node gain for exclude node 2 (n_{10}^2) : n_{10}^2=β_1+β_2-$\Delta\beta_{10}^2$=4+1-2=3

where $\beta_{10,8}$ is the number of receiving node involved from source node 10 to node 8, β_1 and β_2 is the respective number of receiving node involved in the transmission range which node 1 and node 2 transmits in MRN-MST.

[7]Node j is said to be a *child* node of node i if node j is included in a broadcast tree by node i

For excluding node 8 :

source node 10 has to increase its number of receiving node : $\Delta\beta_{10}^{8}=\beta_{10,5}$-$\beta_{10}$=9-6=3
receiving node gain for excluding node 8 (n_{10}^{8}) : $n_{10}^{8}=\beta_1+\beta_2+\beta_8$-$\Delta\beta_{10}^{8}$=4+1+2-3=4

where $\beta_{10,5}$ is the number of receiving node involved from source node 10 to node 5, β_1, β_2 and β_2 is the respective number of receiving node involved in the transmission range which node 1, node 2 and node 8 transmits in MRN-MST.

For the receiving node gain of excluding node 8 is higher, we select to exclude node 8. Thus the source node 10 is selected in the set F to transmit with energy that maximizes its gain, that is $e_{10}=\omega_{10,5}$=34 at which it can cover nodes 1, 2, 8 and all their *child* nodes in MRN-MST.

Continue, we repeat the procedure until all the nodes are covered by the new broadcast tree.

In the example, after the last steps, we get C={1, 2, 3, 4, 5, 6, 7, 8, 9, 10 }, E={1, 2, 8} and F={10}. All nodes are covered(C={1, 2, 3, 4, 5, 6, 7, 8, 9, 10 }) and the algorithm stop.

Finally, we get E={1, 2, 8}, F={10}. which the sum of the transmission powers at each of the transmission nodes consisted the total power required to maintain this tree. The result tree, shown in Figure 31. We can get the total transmission power of the final broadcast tree :

$$e_{MRN\text{-}EWMA}=e_{10}=34$$

We also can get the total receiving node of the final broadcast tree :

$$\beta_{EWMA}=\beta_{10}=9.$$

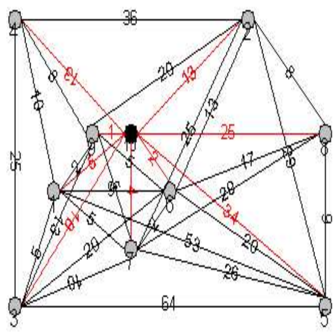


FIG. 31 – The broadcast tree obtained by MRN-EWMA–$e_{MRN\text{-}EWMA}$=34

3.2 Minimum Receiving Node Broadcast Incremental Power

This algorithm is based on the algorithm BIP which is introduced by Wieselthier et al.[17]. It constructs the tree by first determining the node that the source can reach with minimum expenditure of power; if there are more than one node that the source can reach with the same minimum expenditure of power, select node which the edge from source node to it involve the minimum receiving node. After the first node has been added to the tree, MRN-BIP continues by determining which uncovered node can be added to the tree at *minimum additional cost*; and also if there are more than one uncovered node have the same *minimum additional cost*, we select the node with *minimum additional receiving node*. Thus at some iteration of MRN-BIP, the nodes that have already included some node in the tree can additionally increase their transmission power to reach some other yet uncovered node but with the minimum receiving node in the transmission range.

The difference between these two algorithms is that the BIP only consider the transmission power when select a new node to the tree ; if there are more than one new node have the same minimum additional cost, the BIP do not afford a regulation for the node's selection, so the selection has to be randomly. But MRN-BIP solve the problem very well, when there are more than one new node have the same minimum additional cost, it will consider their additional receiving node and select the node with minimum additional receiving node to the tree. With this standard, it will give a guarantee of constructing a minimum transmission and receive energy consumption broadcast tree.

In order to compare convenient, we use the same 10 node network example (see Figure 4) by using algorithm MRN-BIP to construct a minimum receiving node and minimum energy broadcast tree. **Initially**, the tree consists of only the source node, we begin by determining the node that the source can reach with minimum expenditure of power and involved minimum receiving node. In the example, we have :

$\omega(10,1)=5$	$\beta(10,1)=4$	$\omega(10,4)=13$	$\beta(10,4)=6$	$\omega(10,7)=4$	$\beta(10,7)=3$
$\omega(10,2)=13$	$\beta(10,2)=6$	$\omega(10,5)=34$	$\beta(10,5)=9$	$\omega(10,8)=25$	$\beta(10,8)=8$
$\omega(10,3)=18$	$\beta(10,3)=7$	$\omega(10,6)=2$	$\beta(10,6)=2$	$\omega(10,9)=1$	$\beta(10,9)=1$

The node 9 is the nearest node of node 10, add it to the tree, see Figure 32. We can get :

$$e_{10}=\omega(10,9)=1,\ \beta_{10}=\beta(10,9)=1$$

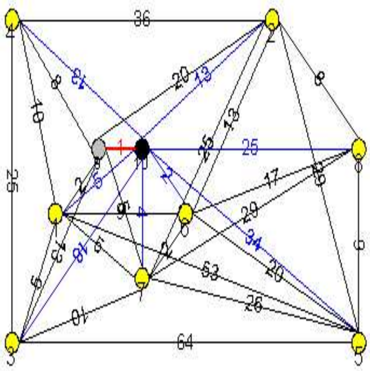


FIG. 32 – The network example of MRN-BIP step1 : 10→9

Second, we determine the node with *minimum additional cost* and *minimum additional receiving node* to add to the tree.

In the example, we can increase the node 10's power to reach a second node or can let node 9 transmit to its nearest neighbor that is not already in the tree. We have :

$$min\Delta e_{10}=\omega(10,6)\text{-}e_{10}=1 \quad min\Delta e_9=\omega(9,1)=2$$

We then add node 6 to the tree, see Figure 33. We can get :

$$e_{10}=\omega(10,6)=2 \quad \beta_{10}=\beta(10,6)=2$$

At this point, there are three nodes in the tree, namely node 10, node 9 and node 6. For each node, we determine the incremental cost and receiving node to reach a new node. We have :

$$\begin{array}{ll} min\Delta e_{10}=\omega(10,7)\text{-}e_{10}=2 & \Delta\beta_{10}=\beta(10,7)\text{-}\beta_{10}=1 \\ min\Delta e_9=\omega(9,1)=2 & \Delta\beta_9=\beta(9,1)=2 \\ min\Delta e_6=\omega(6,7)=2 & \Delta\beta_6=\beta(6,7)=2 \end{array}$$

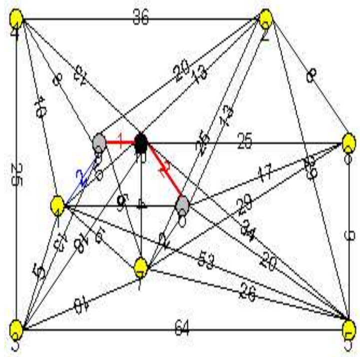


FIG. 33 – The network example of MRN-BIP step2 : 10→6

Node 10 reach node 7, node 9 reach node 1 and node 6 reach node 7 have the same additional cost, so we have to consider the additional receiving node of these three edges. We find that node 10 reach node 7 will get the minimum additional cost and minimum additional receiving node, we add node 7 to the tree, see Figure 34. We can get :

$$e_{10}=\omega(10,7)=4 \quad \beta_{10}=\beta(10,7)=3$$

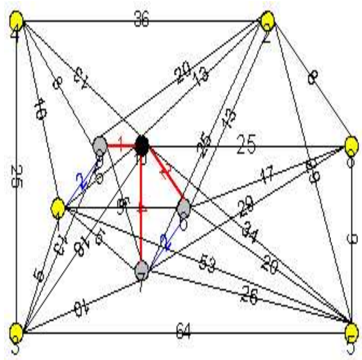
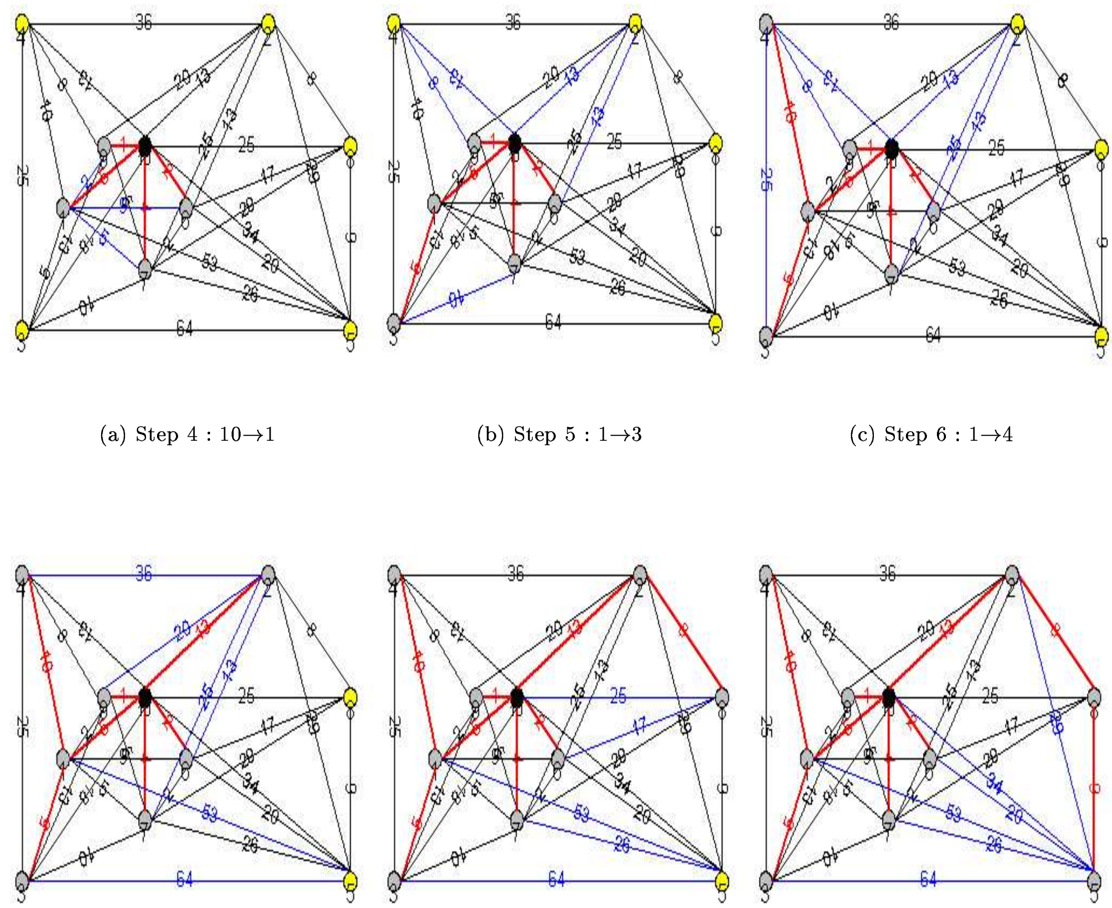


FIG. 34 – The network example of MRN-BIP step3 : 10→7

Continue, we continue the procedure until all the nodes are included in the tree, as shown in Figure 35. The algorithm terminate.



(a) Step 4 : 10→1 (b) Step 5 : 1→3 (c) Step 6 : 1→4

(d) Step 7 : 10→2 (e) Step 8 : 2→8 (f) Step 9 : 8→5

FIG. 35 – The example for algorithm MRN-BIP step4-9

Finally, we get the final minimum receiving node and minimum energy broadcast tree, see Figure 36. In the final tree, the transmission nodes are node 1, node 2, node 8 and node 10. We can get :

$$e_{10}=13 \quad \beta_{10}=6 \qquad e_8=9 \quad \beta_8=2$$
$$e_2=8 \quad \beta_8=1 \qquad e_1=10 \quad \beta_1=5$$

The total transmission power of the broadcast tree is :

$$e_{MRN-BIP}=e_1+e_2+e_8+e_{10}=35$$

The total receiving node of the broadcast tree is :

$$\beta_{MRN-BIP}=\beta_1+\beta_2+\beta_8+\beta_{10}=13$$

Now we have introduced two algorithms for constructing the minimum receiving node minimum energy broadcast tree in *all-wireless networks*, we will compare the results of these heuristics and the origins in the next subsection and give the result of the performance evaluation for our two heuristics.

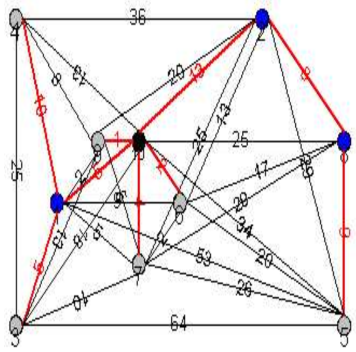


FIG. 36 – Final tree obtained by algorithm MRN-BIP–$e_{MRN-BIP}$=35

3.3 Result

Based on the broadcast tree constructed respective by our two heuristics and the proposed origins, we can get the result as following :

Table 3.1 : Compare of heuristics and the origins

	BIP	**MRN-BIP**	**EWMA**	**MRN-EWMA**
Transmission power	34	35	34	34
Number of receiving node	16	12	10	9

We get the result of the two heuristics, for this network example, which is better than the origins. But we have to do the performance evaluation to evaluate them.

4 PERFORMANCE EVALUATION

We performed a simulation study to evaluate our centralized algorithms (MRN-EWMA and MRN-BIP) version.

We compared the centralized version of our algorithms (MRN-EWMA and MRN-BIP) with EWMA and BIP algorithms. The simulations were performed using networks of eight different sizes : 10, 20, 30, 40, 50, 60, 70 and 100 nodes. The nodes in the networks are distributed according to a spatial Poisson distribution over the same deployment region. Thus, the higher the number of nodes the higher the network density. The source node for each simulation is selected randomly from the overall set of nodes. The maximum transmission range is chosen such that each node can reach all other nodes in the network. The transmission power used by a node in transmission depends on the reached distance. Similarly to Wieselthier et al. in [17], and Čagalj et al. in [16] , we run 100 simulations for each simulation setup consisting of a network of a specified size and a algorithm. The performance metric used is the total transmission power and the total receiving node of the broadcast tree. Here we use the ideas of the *normalized tree transmission power* [17] and the *normalized tree receiving node.*

Let $P_i(m)$ denote the total power of the broadcast tree for a network instance m, generated by algorithm i (i={EWMA, MRN-EWMA, BIP, MRN-BIP}). Let $P_0(m)$ be the power of the lowest-power broadcast tree among the set of algorithms performed and particular network instance m. Then the normalized tree transmission power associated with algorithm i and network instance m is defined as follows : $P_i^{'}(m) = \frac{P_i(m)}{P_0}$. The $P_i^{'}(m)$ provide a measure how close each algorithm comes to providing the lowest-power tree.

Let $N_i(m)$ denote the total receiving node of the broadcast tree for a network instance m, generated by algorithm i (i={EWMA, MRN-EWMA, BIP, MRN-BIP}). Let $N_0(m)$ be the receiving node of the least-receiving node broadcast tree among the set of algorithms performed and particular network instance m. Then the normalized tree receiving node associated with algorithm i and network instance m is defined as follows : $N_i^{'}(m)=\frac{N_i(m)}{N_0}$. The $N_i^{'}(m)$ provide a measure how close each algorithm comes to providing the less-receiving node tree.

Let us first consider the performance of the algorithms shown in Figure 37. In the figure 37, we can see the average normalized tree transmission power (shown on the vertical axis) achieved by the algorithms on networks of different sizes (the horizontal axis). The figure shows that the solutions for the broadcast tree obtained by MRN-EWMA have, on the average, almost equal to the solution of EWMA but a little lower-power than it ; the solutions for the broadcast tree obtained by MRN-BIP have, on the average, almost lower-power than the solution of BIP, even much lower-cost in the big network size (100 nodes in our case).

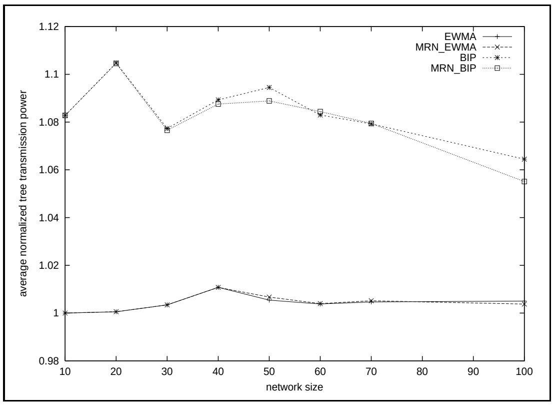


FIG. 37 – Normalized tree transmission power for 100 network instances

However, we can notice that for the receiving node, the figure 38 shows that the average normalized tree receiving node achieved by the algorithms on networks of different sizes. We can get that the solution for the broadcast tree obtained by MRN-EWMA have, on the average, less receiving nodes than the solution of EWMA ; the solutions for the broadcast tree obtained by MRN-BIP have, on the average, much less receiving nodes than the solution of BIP.

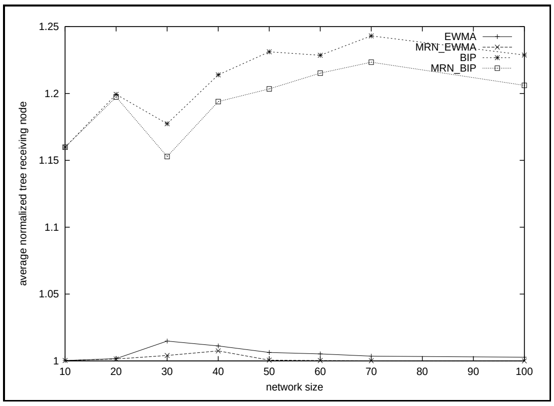


FIG. 38 – Normalized tree receiving node for 100 network instances

Our algorithms are based on the origins with some adaptions to make the origins more precious, the main functions are same with the origins. We can get the transmission power consumption between the adapted version and the origin is almost same but the former has the less receiving node. When the network size increased, the opportunity of random selections for the origins will increase, relatively the different selections for the adaption version will be more, so the receiving node difference will be enlarged. That is why we get the same result in small network size but get bigger difference in large network size.

Based on our simulation results, we conclude that MRN-EWMA and MRN-BIP utilizes the wireless multicast advantage property at least as well as EWMA and BIP, but lower receiving energy consumption than them. Especially in big size network, the difference is obviously. For this reasons, we think MRN-EWMA and MRN-BIP to be preferable to EWMA and BIP.

5 CONCLUSIONS

We have provided novel contributions on the two most relevant aspects of transmission power-efficient and receiving node-efficient broadcast in all-wireless networks. First, we studied the minimum receiving node and minimum energy problem in unicast environment. We discussed the algorithm MRN-SPT, which to construct a minimum receiving node shortest path tree, and proved to have a better performance than the origin SPT algorithm in *all-wireless* networks. Then we provided a proof for this algorithm MRN-SPT.

Second, we introduced three algorithms for the transmission energy efficient broadcast in *all-wireless* networks : EWMA, BIP MINI_BROAD_TREE. And based on these algorithms, we provided two heuristics called Minimum Receiving Node Embedded Wireless Multicast Advantage (MRN-EWMA) and Minimum Receiving Node Broadcast Incremental Power (MRN-BIP). We show that these algorithm act more prominent than the original proposal provided in the literature : EWMA and BIP.

In terms of future work, we intend to explore how our heuristics can be distributed which other authors have reckoned to be both necessary and challenging. Moreover, we will explore how to extend our proposal to multicast. Finally, we intend to study how to cope with the mobility of the nodes.

6 ACKNOWLEDGMENTS

We thank Madam Johanne Cohen for helpful directions, forceful support and proper comments. And thanks also all the members of the jury for a kindly attention to this paper.

Bibliographie

[1] Julien Cartigny, David Simplot, and Ivan Stojmentović. Localized minimum-energy broadcasting in ad-hoc networks. 2001.

[2] Moses Charikar, Chandra Chekuri, To yat Cheung, Zuo Dai, Ashish Goel, Sudipto Guha, and Ming Li. Approximation algorithms for directed steiner problems. *J. Algorithms*, 1999.

[3] T.H. Corman, C.E. Leiserson, and R.L. Riviest. *Introduction to algorithms*. 1990.

[4] Arindam K. Das, Robert J. Marks, Mohamed El-Sharkawi, Payman Arabshahi, and Andrew Gray. e-merge : A heuristic for improving minimum power broadcast trees in wireless networks. 2002.

[5] Laura Marie Feeney. Energy efficinet communication in ad hoc wireless network. 2001.

[6] Javier Gomez, Andrew T. Campbell, Mahmoud Naghshineh, and Chatschik Bisdikian. Paro : Supporting dynamic power controlled routing in wireless ad hoc networks. 2001.

[7] Zygmunt J. Haas, Jing Deng, Ben Liang, Panagiotis Papadimitratos, and S. Sajama. Wireless ad hoc networks. 2002.

[8] Christine E. Jones, Krishna M. Sivalingam, Prathima Agrawal, and Jyh Cheng Chen. A survey of energy efficient network protocols for wireless networks. *Wireless Networks 2001*, 7 :343–358, 2001.

[9] Fulu Li and Ioanis Nikolaidis. On minimum-energy broadcasting in all-wireless networks. *Proceedings of the 26th Annual IEEE Conference on Local Computer Networks (LCN 2001)*, 2001.

[10] Weifa Liang. Constructing minimum-energy broadcast trees in wireless ad hoc networks. *MOBIHOC'02*, 2002.

[11] Guolong Lin and Guevara Noubir. Energy efficient broadcast with sectored antennas in wireless ad hoc networks. *IASTED/IEEE Wireless and Optical Communications Conference*, 2002.

[12] Paul Mühlethaler. *802.11 et les réseaux sans fil*. Eyrolles, 2002.

[13] Aleksi Penttinen. Research on ad hoc networking : Current activity and future directions. 2001.

[14] Charles E. Perkins. *Ad Hoc Networking*. Addison Wesley, 2001.

[15] Anan Phonphoem. Ad hoc wireless networks. 2001.

[16] Mario Čagalj, Jean-Pierre Hubaux, and Christian Enz. Minimum-energy broadcast in all-wireless networks : Np-completeness and distribution issues. *MOBICOM'02*, 2002.

[17] Jeffrey E. Wieselthier, Gam D. Nguyen, and Anthony Ephremides. On the construction of energy-efficient broadcast and multicast trees in wireless networks. *IEEE INFOCOM 2000*, pages 586–594, 2000.

[18] Jeffrey E. Wieselthier, Gam D. Nguyen, and Anthony Ephremides. Energy-efficient broadcast and multicast trees in wireless networks. *Mobile Networks and Application*, 7 :481–492, 2002.